Dimension Reduction Techniques and the Classification of Bent Double Galaxies

Imola K. Fodor and Chandrika Kamath

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-560
Livermore CA 94551

Abstract

As data mining gains acceptance in the analysis of massive data sets, it is becoming clear that we need algorithms that can handle not only the massive size, but also the high dimensionality of the data. Certain pattern recognition algorithms can become computationally intractable when the number of features reaches hundreds or even thousands, while others break down if there are large correlations among the features. A common solution to these problems is to reduce the dimension, either in conjunction with the pattern recognition algorithm or independent of it.

In this paper, we describe how dimension reduction techniques can help in the classification of radio galaxies with a bent double morphology. We consider decision tree and generalized linear model classifiers, and explain the statistical and exploratory data analysis methods we use to address the problem of high dimensionality by selecting the features that are relevant to the problem. We show that a careful extraction and selection of features is necessary for the successful application of data mining techniques.

Key words: Data mining, exploratory data analysis, feature selection, dimension reduction, classification, decision trees, generalized linear models

1 Introduction

As commercial and scientific datasets approach the terabyte and even petabyte range, it is no longer possible to manually find useful information in such data.

Email address: `fodor1@llnl.gov and kamath2@llnl.gov` (Imola K. Fodor and Chandrika Kamath).

To address this problem, semi-automated techniques from data mining are increasingly being used as a viable means of analyzing these massive data sets. Data mining is an iterative and interactive process, involving pre-processing of the data, the search for patterns, and the interpretation and validation of the results. In data pre-processing, relevant high-level features (also called attributes or variables in different fields) are extracted from the low level data, and in pattern recognition, a pattern in the data is recognized using these features.

In many problems, the number of features extracted from the data may be quite large, numbering in the hundreds or even thousands. This can make the task of pattern recognition difficult and time consuming as many of the features extracted may be irrelevant to the problem being addressed. Possible high correlations among the features may render certain pattern recognition methods invalid. In order to reduce the number of features to a more manageable, well-selected set most relevant to the problem, dimension reduction techniques are often used. These techniques can be applied either in conjunction with the pattern recognition task, or independent of it. They may or may not exploit information available from the problem domain and the problem itself. We note that dimension reduction, dimensionality reduction, feature selection, and variable selection, are all similar terms used by different communities.

In this paper, we focus on dimension reduction techniques as they are applied in the context of detecting radio-emitting galaxies with a bent double morphology in the FIRST astronomical survey. Our objective is to show that a careful extraction and selection of features is necessary for the success of any data mining endeavor.

The paper is organized as follows. After a brief overview of data mining in Section 2, we discuss the important role dimension reduction plays in the accurate and efficient identification of patterns in the data. Then, in Section 3, we briefly describe the problem we are solving using data mining techniques, that is, the classification of bent double galaxies. Next, in Section 4, we discuss the techniques commonly used in dimension reduction, focusing on those applicable in the context of our problem. This is followed in Section 5 by an overview of the classification techniques used in the identification of bent double galaxies. Our experimental results are presented in Section 6, followed by our conclusions in Section 7.

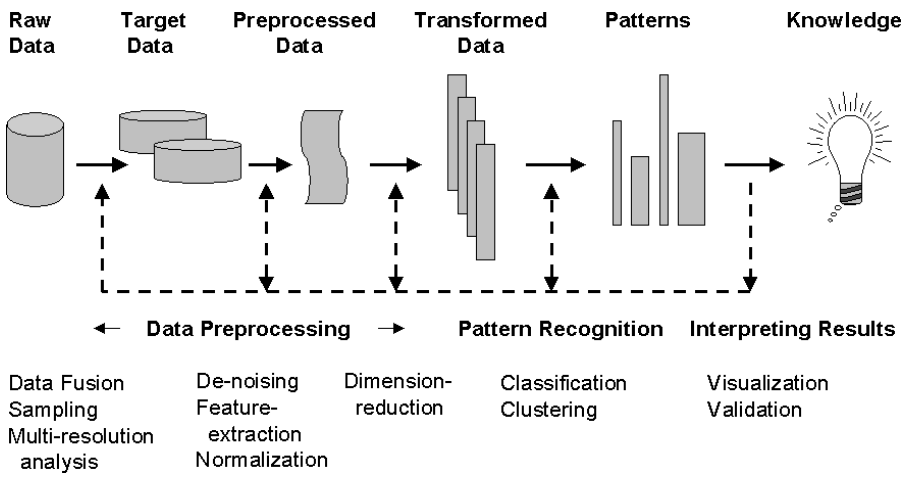


Fig. 1. Data mining: an interactive and iterative process.

2 Introduction to Data Mining

Data mining is a process concerned with uncovering patterns, associations, anomalies and statistically significant structures in data [8,15,24]. It typically refers to the case where the data is too large or too complex to allow either a manual analysis or analysis by means of simple queries. As illustrated in Figure 1, data mining consists of two main steps, data pre-processing and pattern recognition. Pre-processing the data is often a time-consuming, yet critical first step. To ensure the success of the data mining process, it is important that the features extracted from the data are relevant to the problem and representative of the data.

Depending on the type of data being mined, the pre-processing step may consist of several sub-tasks. These may include sampling to reduce the number of instances, multi-resolution techniques to coarsen the resolution of the data, data fusion to exploit data from different sources, de-noising of the data, extraction of features from the data, and the normalization of the features. At the end of this first step, we have a feature vector for each data instance. Depending on the problem and the data, we may need to reduce the number of features using feature selection or dimension reduction techniques such as principal component analysis or its non-linear extensions. The data is now ready for the identification of patterns through the use of algorithms such as classification, clustering, and regression. These patterns are then displayed to the user for validation. Data mining is an iterative and interactive process. The output of any step, or feedback from the domain experts, could result in an iterative refinement of any, or all, of the sub-tasks.

Data mining techniques are being applied for the analysis of data in a variety of fields including remote sensing, bio-informatics, medical imaging, astronomy, web mining, text mining, customer relationship management, and market-

basket analysis. While much of the focus tends to be on the pattern recognition algorithms, it is the data pre-processing tasks that are more influential in the success of the data mining endeavor [17,4]. Many of these tasks are also domain- and problem-dependent, making a general discussion of them difficult.

In this paper, we focus on one of the tasks in data pre-processing, namely, dimension reduction, or the reduction in the number of features that are used to represent an object. There are several reasons why this is an important task. First, the computation required in the pattern recognition task that follows the extraction of features is dependent on the number of features that represent an object. For example, the creation of a decision tree classifier requires $O(mn\log(n))$ operations, where n is the number of instances, m is the number of features, and the tree is assumed to be of depth $\log(n)$. Therefore, if the number of features is large, more computations are performed, and, if these features are not discriminatory, they only increase the computations performed. Second, experiments have shown that adding a random binary feature to a standard dataset can cause the accuracy of a decision tree classifier to degrade by 5% to 10%. This is because at some point in the creation of the decision tree, the data available to select the decision variable is small enough that the random feature is selected for the split on that node of the tree. This results in random errors when the tree is used for classification. Similar results are also observed with other classifiers [29]. Third, some classification methods, such as the ones based on linear models, break down if there are correlations among the features. Lastly, in many pattern recognition tasks, the number of features represents the dimensions of a search space - the larger the number of features, the greater the dimension of the search space, and the harder the problem. The resulting curse of dimension, as well as its blessings are discussed in greater detail in [7]. In light of these observations, dimension reduction or feature selection techniques are often applied prior to the pattern recognition step in data mining.

3 Searching for Bent Doubles in the FIRST Survey

In this paper, we describe how we are using dimension reduction techniques in the process of mining the FIRST astronomical survey. The Faint Images of the Radio Sky at Twenty-cm (FIRST) survey [1] is producing the radio equivalent of the Palomar Observatory Sky Survey, and when complete, will cover more than 10,000 square degrees of the sky to a flux density limit of 1.0 mJy (milli-Jansky). The data collected through 1999 has covered about 8,000 square degrees, producing more than 32,000 two-million pixel images. At a threshold of 1.0 mJy, there are approximately 90 radio-emitting galaxies, or radio sources, in a typical square degree.

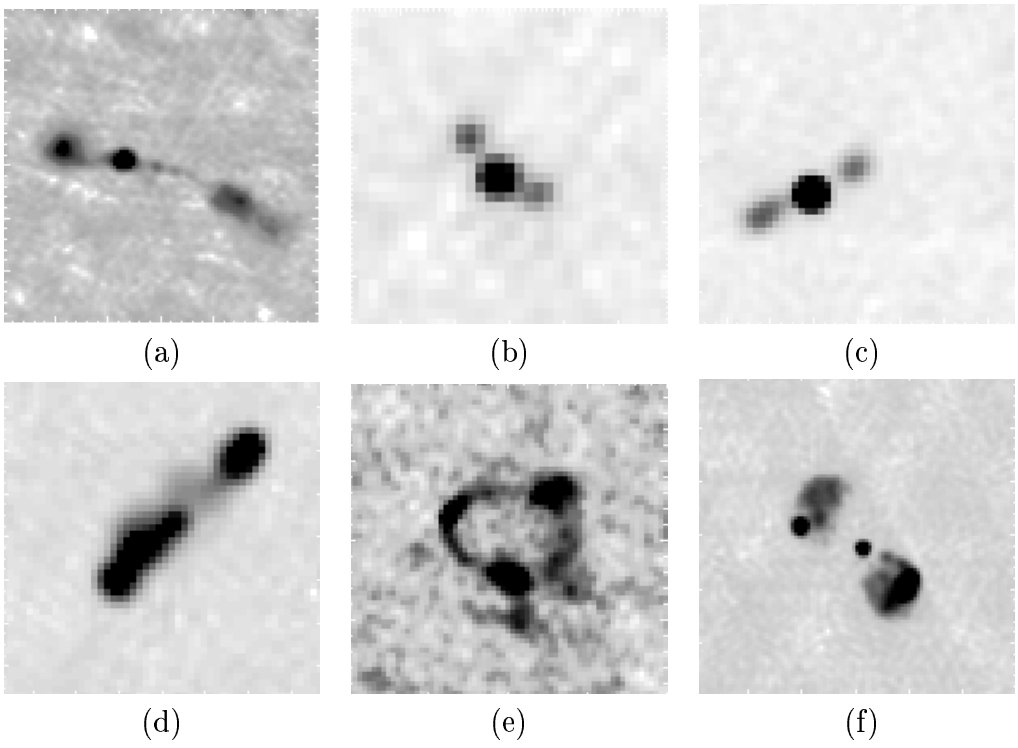
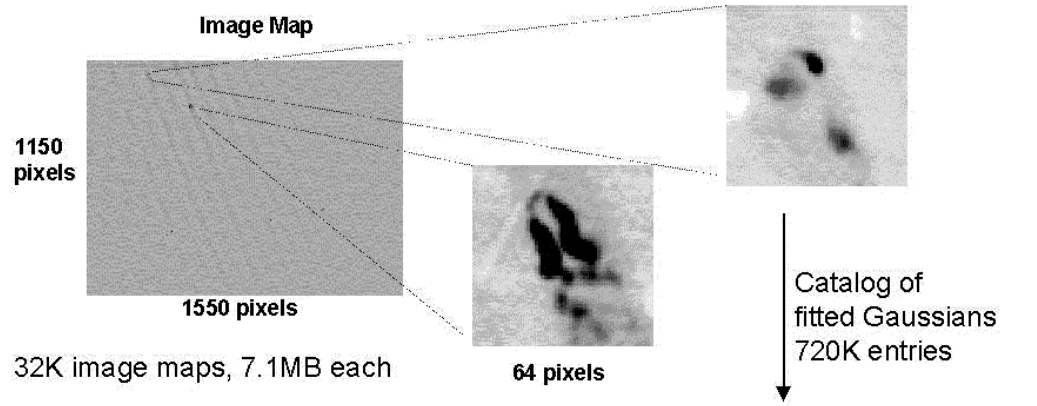


Fig. 2. Example radio sources from FIRST: (a)-(b) bent doubles, (c)-(d) non-bent doubles, (e)-(f) complex sources.

Radio sources exhibit a wide range of morphological types (Figure 2) that provide clues to the source class, emission mechanism, and properties of the surrounding medium. Of particular interest are sources with a bent double morphology as they indicate the presence of large clusters of galaxies. Currently, FIRST scientists identify a bent double through a visual inspection, a process that is not only subjective, but also tedious, especially as the complete survey will have nearly a million galaxies. Our goal is to replace this manual classification by a more automated one, using techniques from data mining.

The data from the FIRST survey, both raw and post-processed, are readily available at their web site (http://sundog.stsci.edu/). A user-friendly interface enables easy access to radio sources at a given RA (Right Ascension, analogous to longitude) and Dec (Declination, analogous to latitude) position in the sky. There are two forms of data available for use — image maps and a catalog. Figure 3 shows an image map containing examples of two bent doubles. These large image maps, totaling about 250 gigabytes, are mostly composed of background noise, with a few pixels corresponding to the radio sources. The source catalog is obtained by processing an image map by fitting two-dimensional elliptic Gaussians to each radio source [28]. For example, the upper bent double in Figure 3 is approximated by the three Gaussians shown in the table in the lower part of the figure. Each entry in the catalog corresponds to the information on a single Gaussian. This includes, among other things, the RA



	RA	DEC	Peak Flux (mJy/bm)	Major Axis (arcsec)	Minor Axis (arcsec)	Position Angle (degrees)
Catalog entry →	00 56 25	-01 15 43	25.38	7.39	2.23	37.9
Radio source	00 56 26	-01 15 57	5.50	18.30	14.29	94.2
	00 56 24	-01 16 31	6.44	19.34	10.19	39.8

Fig. 3. An example FIRST image and catalog detail.

and Dec for the center of the Gaussian, the peak flux, the major and minor axes, and the position angle of the major axis. The catalog is much smaller at 78 Megabytes, but contains information about parts of a radio source, rather than the whole radio source.

In our analysis, we decided to first focus on the data from the catalog. It was not only smaller, but, according to the astronomers, a good approximation to all but the most complex of radio sources. Our first step was to group the catalog entries into radio sources, based on their distances from each other. Next, we separated the data depending on the number of catalog entries comprising each radio source. We have a data set each for all the 1-entry, the 2-entry, the 3-entry, and the 3-plus-entry sources. Then, we excluded the single-entry sources on the assumption that they were unlikely to be bent doubles. We also flagged all the 3-plus-entry sources as they were often complex and likely to be of interest to astronomers. This left us with radio sources with 2 and 3 entries. As the number of features extracted depended on the number of catalog entries, and we wanted feature vectors of equal length, we worked with the 2- and 3-entry sources separately. However, this also meant that the original training set was now divided into two smaller training sets. This "training" set was composed of the few galaxies already identified as bent doubles and non-bent doubles by the astronomers.

Next, we obtained the features, as indicated in Section 6.1. Since the number of features was quite large relative to the size of the training set, in Section 6.2 we applied the dimension reduction techniques described in Section 4. Then, in Section 6.3, we experimented with the classification algorithms described in Section 5. The accuracy of the classifiers was evaluated using cross-validation

methods. If the accuracy was not sufficient, additional features were calculated based on the misclassified instances. The process was repeated until the accuracy was acceptable to the astronomers. Finally, we applied the "best" classifiers to the rest of the survey, and classified the unlabeled galaxies. Additional details on our approach, as well as the problems encountered in mining the FIRST survey for bent double galaxies, are discussed in [9,14].

4 Techniques for Dimension Reduction

As we have described earlier, once the features have been extracted in the data mining process, their number must be reduced to make the task of pattern recognition tractable. There are several ways in which this can be done. Some techniques exploit domain knowledge, others do not. Some techniques are coupled with the task performed in pattern recognition, while others are not.

In many problems, domain knowledge and common sense can be a simple first approach to reducing the number of features. For example, in the classification of bent double galaxies, it is unlikely that the position of the galaxy in the sky (i.e. RA and Dec) is a relevant feature. However, for other features such as the relative spatial distances between the catalog entries or the maximum intensity of a catalog entry, it may be difficult for the scientists to determine if the feature is relevant to the problem at hand or not.

For such problems, several different approaches have been proposed. In the machine learning community, a common approach to identifying the most relevant features is through the use of wrappers [16]. Here the feature selection is done using the classification algorithm as a black box to evaluate the selection. Starting with a training set for the classification problem, and an initial set of features, various subsets of these features are selected. For each subset, the algorithm is used to generate a model based on the training data. The performance of a subset is measured by how well it classifies the test set. The best feature subset found is then used in the actual design of the classification system. A different approach to feature selection is described in [2], where the authors combine feature selection with pattern recognition. They consider the binary classification problem of discriminating between two given sets in an n-dimensional feature space by using as few of the given features as possible. This problem is converted to a mathematical programming problem, with a parametric objective function that achieves the task by generating a separating plane in a feature space of as small a dimension as possible, while minimizing the average distance of misclassified points to the plane.

For our work in the classification of bent double galaxies, we have used the more traditional approaches to feature selection that are based on statistics.

These include include exploratory data analysis (EDA) and principal component analysis (PCA). We describe these in further detail below.

4.1 Exploratory Data Analysis

Exploratory data analysis (EDA) [27,11] consists of a suite of simple techniques that probe a dataset and can be used to understand the data and the relationships among the features. The techniques include simple summary statistics, pairwise plots, box-plots and parallel coordinate plots.

For high dimensional problems, it is often difficult to efficiently visualize the data. As a result, standard multivariate plotting techniques, such as pairwise plots and 3-D plots of three variables at a time, are of limited use in extracting the complicated multivariate relationships among the features. Nevertheless, we resorted to those techniques as means to explore different aspects of the data. In addition, we also explored parallel coordinate plots, where, instead of the perpendicular coordinate system, the coordinates are parallel [12]. This allowed us to visualize a nearly unlimited number of features in a single plot, instead of the maximum of three with the perpendicular coordinate system. We do note that there are ways of including more than three variables in the traditional methods, for example, by using arrows of different directions, magnitudes and colors [10] for the different features, or the cartoon faces of Chernoff to represent up to 15 variables [5]. Since we find it hard to interpret such plots, we have not included them in our analysis.

4.2 Principal Component Analysis

The principal components (PCs) of a multivariate dataset are mutually orthogonal linear combinations of the variables in the original dataset, such that the first PC has the largest variance, the second PC the second largest variance, and so on [13]. Since the variance depends on the scale of the variables, the variables are generally standardized to have mean zero and variance one before calculating the PCs. Otherwise, the variables with the largest absolute variances will dominate. After standardization, all the variables are on the same scale, so that the largest relative variances dominate. The PCs are the eigenvectors of the data covariance matrix, with the first PC being the eigenvector corresponding to the largest eigenvalue. For many datasets, the first few PCs capture most of the variability, and thus provide a compact representation of the important features in the data.

5 Classification Techniques

Classification is a pattern recognition technique in which an algorithm learns a function that maps a data item into one of several pre-defined classes. These algorithms typically have two phases. In the training phase, the algorithm is "trained" by presenting it with a set of examples with known classification. In the test phase, the model created in the training phase is tested to determine how well it classifies known examples. If the results meet expected accuracy, the model is put into operation to classify examples with unknown classification. In this section, we briefly describe the two classifiers we are using for the classification of bent doubles: decision trees and generalized linear models (GLM).

5.1 Decision Tree Classifiers

A decision tree [3,22,23] is a structure that is either a leaf, indicating a class, or a decision node that specifies some test to be carried out on a feature (or a combination of features), with a branch and sub-tree for each possible outcome of the test. The decision at each node of the tree is made to reveal the structure in the data. Decision trees tend to be relatively simple to implement and yield results that can be interpreted easily.

The feature to test at each node of the tree, as well as the value against which to test it, can be determined using one of several measures [21]. Depending on whether the measure evaluates the goodness or badness of a split, it can be either maximized or minimized. In our work, we use the C5.0 decision tree software (Rulequest Research, http://www.rulequest.com), which uses the information gain as the criterion to determine the split. Let T be the set of n examples at a node that belong to one of k classes, and T_L and T_R be the two non-overlapping subsets that result from the split (that is, the left and right subsets). Let L_j and R_j be the number of instances of class j on the left and the right, respectively. Then, the information gain associated with a feature is the expected reduction in entropy caused by partitioning the examples according to the feature. Here the entropy characterizes the (im)purity of an arbitrary collection of examples. For example, the entropy prior to the split in our example would be:

$$\text{Entropy}(T) = \sum_{i=1}^{k} -p_i \log_2 p_i \qquad \text{where} \qquad p_i = (L_i + R_i)/n$$

where p_i is the proportion of T belonging to class i and $(L_i + R_i)$ is the number of examples in class i in T. The information gain of a split $S = \{T_L, T_R\}$

relative to T is then given by

$$\text{Gain}(T, S) = \text{Entropy}(T) - \frac{|T_L|}{|T|} * \text{Entropy}(T_L) - \frac{|T_R|}{|T|} * \text{Entropy}(T_R)$$

where T_L and T_R are the subsets of T that correspond to the left and right branches, respectively.

5.2 Generalized Linear Model Classifiers

Linear models [25] explain response variables in terms of linear combinations of explanatory variables. Following standard notation in the statistical literature, given n observations (examples) of the p explanatory variables $\mathbf{x}_i = \{1, x_{i,1}, \cdots, x_{i,p-1}\}$ and the associated response values y_i, the linear model (LM) has the form

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p-1} \\ 1 & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}, \tag{3}$$

where $\boldsymbol{\beta}^T = \{\beta_0, \beta_1, \cdots, \beta_{p-1}\}$ is the unknown regression coefficient vector, and the errors $\boldsymbol{\epsilon}^T = \{\epsilon_0, \epsilon_1, \cdots, \epsilon_n\}$ are assumed to be independent of the explanatory variables. In matrix notation, we can write (3) as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \text{E}(\boldsymbol{\epsilon}) = \mathbf{0}, \quad \text{Cov}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}, \tag{4}$$

where we also indicated that the errors are assumed to have mean zero and positive definite covariance matrix $\boldsymbol{\Sigma}$.

The parameter estimates $\hat{\boldsymbol{\beta}}$ for the LM in (4) are generally obtained either by minimizing the residual sum of squares, or by maximizing the joint likelihood of the observations under a multivariate normal error distribution. Both of these methods result in

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\}. \tag{5}$$

Differentiating with respect to $\boldsymbol{\beta}$ leads to the well-known normal equations

$$\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{y}, \tag{6}$$

and, assuming the inverse exists, to the generalized least-squares (GLS) estimates

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{y}. \tag{7}$$

If $\boldsymbol{\Sigma} = \mathbf{I}_n$, the GLS estimates are called ordinary least squares estimates (OLS); if $\boldsymbol{\Sigma}$ is diagonal, they are called weighted least squares (WLS) estimates.

In our context, the response is a binary variable indicating bent/non-bent, and the galaxy features are the explanatory variables. However, under the LM, there are no restrictions on the fitted values $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and on predicted values $\hat{y}_k = \mathbf{x}_k\hat{\boldsymbol{\beta}}$ corresponding to previously unseen values $\mathbf{x}_k$. In order to model binary responses, we need to guarantee that $\hat{\mathbf{y}}$ and $\hat{y}_k$ are binary.

The generalized linear model (GLM) [20,6] extends the LM by allowing the response variable to be restricted to a certain range, and to have variances that depend on the mean. The GLM models a function of the expected value of y_i in terms of linear combinations of the explanatory variables. Let μ_i denote the expected value of y_i, that is, $\mu_i = \mathrm{E}(y_i)$. Then, a GLM can be written as

$$g(\mu_i) = \mathbf{x}_i\boldsymbol{\beta} \equiv \eta_i \quad \text{and} \quad \mathrm{Var}(y_i) = \phi V(\mu_i), \tag{8}$$

where the monotone increasing link function g describes how the mean depends on the linear predictors, the variance function V specifies how the variance depends on the mean, and ϕ is a constant dispersion parameter.

Binary responses, taking discrete values in $\{0, 1\}$ according to the probability distribution $d(y_i; \mu_i) = \mu_i^{y_i}(1-\mu_i)^{1-y_i}$, fit naturally in the GLM framework. The logit link

$$g(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_i}\right) = \eta_i, \tag{9}$$

or its equivalent inverse link

$$f(\eta_i) = \frac{e^{\eta_i}}{1+e^{\eta_i}} = \mu_i, \tag{10}$$

together with the variance function

$$V(\mu_i) = \mu_i(1-\mu_i) \tag{11}$$

capture the characteristics of binary responses. By Eq. (10), μ_i is in the interval $[0, 1]$, and the variance is accurately described by Eq. (11). We will use this GLM, also known as the logistic regression model, to classify the radio galaxies.

The parameters of a GLM are usually estimated by maximizing the likelihood function corresponding to the assumed model. Unlike in the LM case displayed in Eq. (6), however, the normal equations for the GLM are nonlinear in the

parameters $\boldsymbol{\beta}$, and therefore the optimization problem has no closed-form solution. In practice, the parameters can be estimated by iteratively re-weighted least squares (IRLS), a variant of the Newton algorithm. For the unfamiliar reader, we include the estimation steps in Appendix A.

If $\hat{\boldsymbol{\beta}}$ denotes the IRLS coefficient estimate for the logistic regression model, the mean value estimate corresponding to observation i is given by

$$\hat{\mu}_i = \frac{\exp\{\mathbf{x}_i\hat{\boldsymbol{\beta}}\}}{1+\exp\{\mathbf{x}_i\hat{\boldsymbol{\beta}}\}}. \tag{12}$$

The fitted values $\hat{y}_i$ of the original binary variables can be obtained by

$$\hat{y}_i = I_{\{\hat{\mu}_i \geq p\}}, \tag{13}$$

where $I_{\{\mathrm{a}\}}$ is the $\{0,1\}$ indicator function corresponding to $\{\mathrm{a} = \mathrm{False}, \mathrm{a} = \mathrm{True}\}$, and the fraction p is generally taken to be $p = 0.5$. Predictions corresponding to future values of the explanatory variables $\mathbf{x}_k$ are formed in a similar fashion.

6 Experimental Results

In this section we present the results of the application of various dimension reduction techniques, in conjunction with the classifiers, for the problem of classification of bent double galaxies in the FIRST survey. Recall that our goal is to find a procedure that will best classify the galaxies into bent doubles and non-bent doubles. We focus on the three-entry galaxies, using a training set of 195 examples with 167 bent doubles and 28 non-bent doubles.

In Section 6.1, we indicate the 103 features we extracted from the FIRST catalog. To avoid any unwanted effect due to the different measurement units, we standardized the feature columns to have mean zero and standard deviation one. Next, we reduced the dimension of the feature set, as 103 features were too many compared to the size of the training set. Finally, we applied the classification algorithms described in Section 5. While the decision trees in Section 5.1 are not severely affected by the number and the relevance of the features, the linear models in Section 5.2 are very sensitive to the quality of the input feature set. They assume that the features are linearly independent, and that the number of features is far less than the number of examples.

An important issue in evaluating the performance of classification algorithms is their accuracy. According to the astronomers, for the bent double problem, a method with about 90% accuracy was adequate, since it was not possible to do better with manual identification. So, the misclassification error, including bents classified as non-bents and non-bents classified as bents, of a good model

should be below 10%. We evaluated the performance of the methods using ten-fold cross-validation with the training set. We divided the data randomly into ten parts, selected a model based on nine parts at-a-time, then evaluated it on the remaining one part.

We also make the following observations about our data. Our training set was relatively small as the galaxies had to be manually labeled by the astronomers. It was also not very accurate as the scientists were often subjective and inconsistent in their labeling. In addition, as we are currently using features from only the catalog, we expect good performance only if the "bentness" of a radio source is adequately represented by these features.

6.1 Features for Bent Doubles

We identified relevant features for the classification of galaxies through extensive conversations with the FIRST astronomers. We found that they placed greater focus on spatial features such as distances and angles. Frequently, they would characterize a bent double as a radio-emitting "core" with one or more additional components at various angles, which were usually wakes left by the core as it moved relative to the Earth. Based on this information, we took some of features directly from the FIRST catalog and derived others from the basic ones in the catalog. We focused on features that were scale, rotation and translation invariant, as the bent double pattern was scale, rotation, and translation invariant. We were also interested in features that are robust, that is, not sensitive to small changes in the data. In addition, we included various book-keeping "features", such as radio source ID, to help us easily map the galaxies in our code to the galaxies in the survey. Appendix A lists all the 103 features we calculated from the FIRST catalog.

6.2 Results of Feature Selection

Having identified and extracted the 103 features that could possibly discriminate between a bent double and a non-bent double, our next task was to reduce this large number to something more manageable and minimize any dependencies and redundancies among the features. Section 6.2.1 reports the results of the EDA methods described in Section 4.1, and Section 6.2.2 reports the results of the PCA methods explained in Section 4.2.

6.2.1 Feature Selection Results with Exploratory Data Analysis

We first used EDA to see if we could identify simple relationships among the features that would help us to reduce their number for the bent double problem. Figure 4 displays an example parallel coordinate plot. The values of the nine features selected are plotted on coordinates that are parallel to each other, with the bent doubles in the left panel and non-bent doubles in the right panel.

We notice several relationships among the features. We first note that there are a few possible outliers in the non-bent dataset, corresponding to the four larger-than-usual values in the **peakFlux**, **CFlux** and **CSNR** features. In addition, the almost parallel lines connecting **ARA** and **BRA** indicate that those variables are very highly correlated, which is to be expected as the RA coordinates of the entries A and B in a galaxy are close to each other. It is also clear that the bent doubles in the training set are situated in three different sections of the sky (note the three disjoint sections connecting **ARA** and **BRA** in the left panel), while the non-bents are concentrated in the first two sections. This relationship may make the sky coordinates appear significant in classifying bent doubles, though domain knowledge would indicate otherwise. We also observe that the **totArea** feature is fairly independent of the RA position in the sky (note that the lines from the disjoint sections in **BRA** map to the same single region of **totArea**, except for a few outliers). Also, there is a high negative correlation (=-.99) between **coreAngl** and **ariAngl**.

In addition to parallel plots, we also used other EDA tools such as the correlation matrix of the features and simple box-plots to identify the features that could discriminate among the bent doubles and the non-bent doubles. We used the following guidelines to reduce the number of features:

- if a feature was dependent on the scale or sensitive to small changes in the data, we ignored it.
- we considered simple box-plots to determine if the feature could be used to distinguish between the bent doubles and the non-bent doubles. Based on Figure 5, for example, we ignored both **ABRelPFlux** and **ABAnglDiff**.
- if several features were highly correlated, we kept only one of them. In such cases, we tried to keep the most astronomically relevant feature. However, we note that this selection process was not strictly objective, as we could have chosen different features on several occasions.

The most important features based on the methods of this section are:

(1) **sumIntFlux**
(2) **totEllipt**
(3) **angleAB**
(4) **angleAC**
(5) **totalBendGeom**
(6) **ariAngl**
(7) **ABAnglSide**
(8) **ACAnglSide**

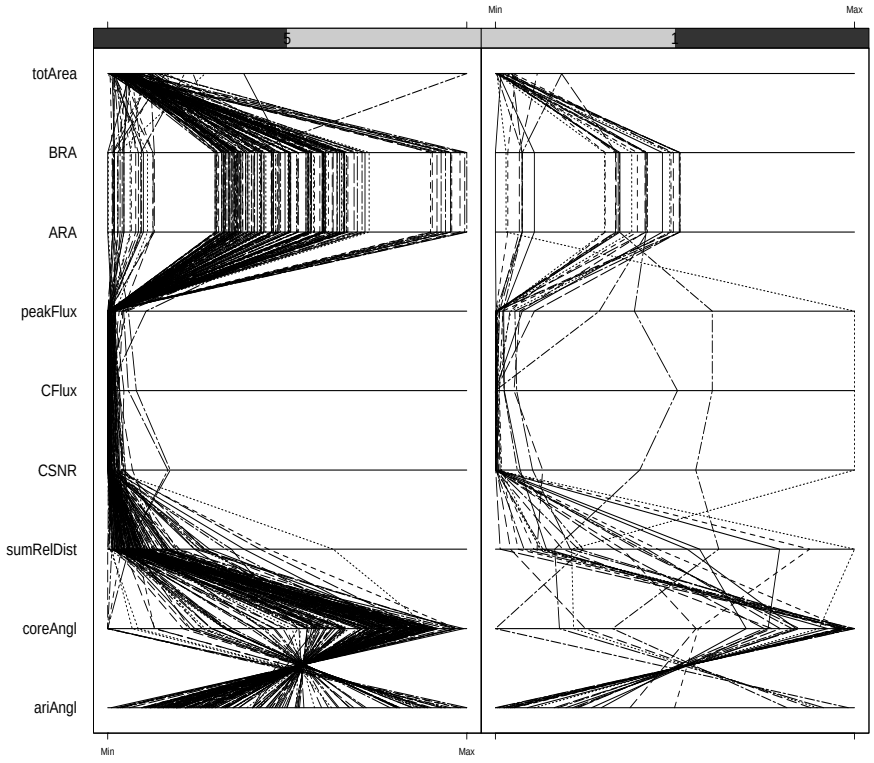


Fig. 4. Example parallel coordinate plots: nine variables split by category of bentness (bent doubles in left panel, non-bent doubles in right panel).

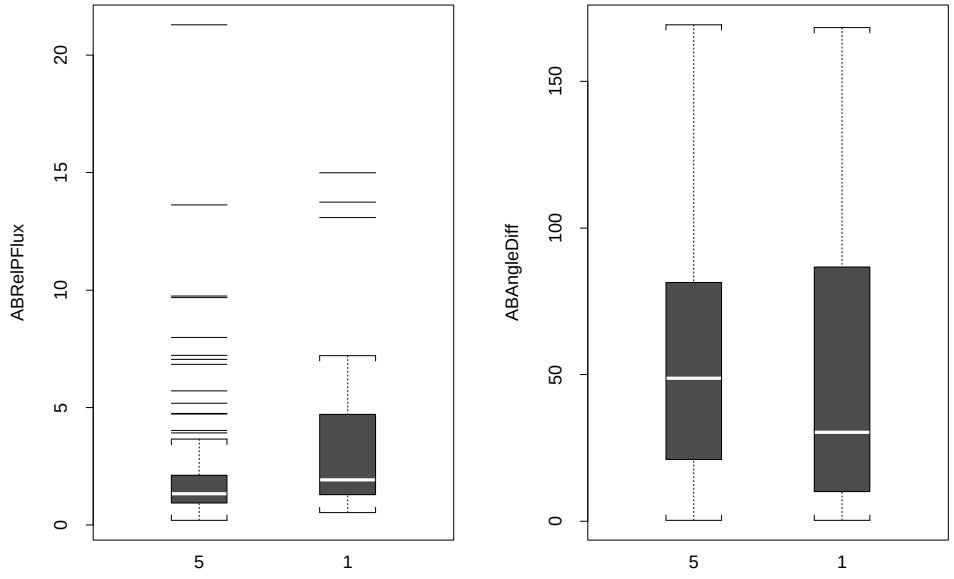


Fig. 5. Example box-plots: pairwise features ABRelPFlux and ABAnglDiff split by bents (5) and non-bents (1).

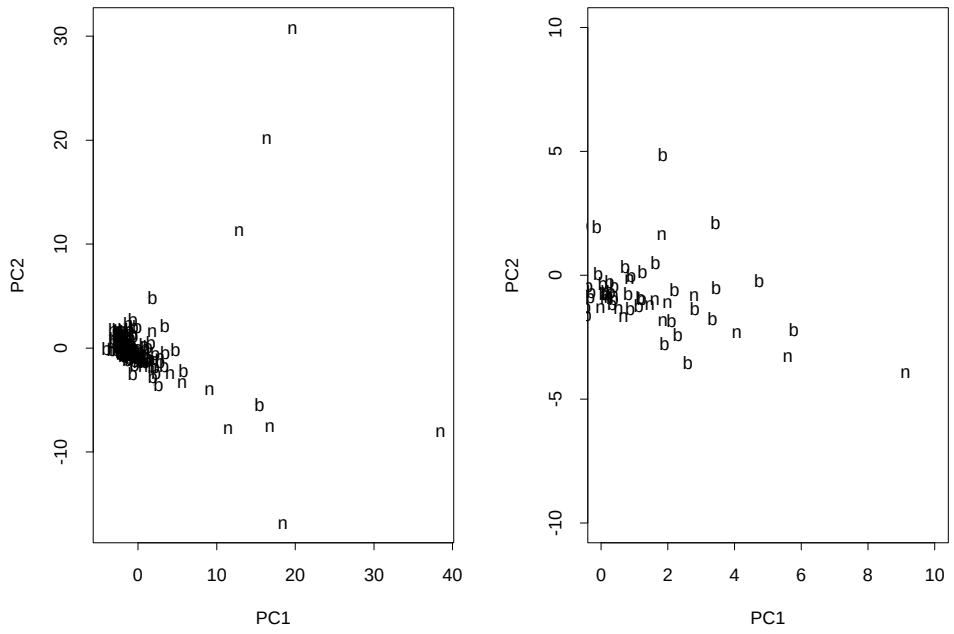


Fig. 6. Bents (b) and non-bents (n) by the first two principal components.

(9) **sumRelDist**
(10) **axialSym**
(11) **ariSym**
(12) **anotherSym**
(13) **pointSrc**
(14) **AB(AC,BC)RelDist**
(15) **A(B,C)Diffusion**
(16) **A(B,C)Ellipt**
(17) **A(B,C)RMS**

6.2.2 Feature Selection Results with PCA

Next, we calculated the PCs of the FIRST features, and found that the first 20 components explained about 90% of the variance. Table 1 presents the individual standard deviations and cumulative percentages corresponding to the first three and to the twenty-th principal component. Figure 6 presents the galaxies in the training set as a function of the first two principal components; the second panel is a zoomed-in version of the first. The bents are labeled by "b" and non-bents by "n" in the two graphs. The first two PCs do not clearly

	PC_1	PC_2	PC_3	...	PC_{20}
Standard deviation	4.5238	3.3435	2.7163	...	0.8727
Proportion of variance	0.2285	0.1248	0.0824	...	0.0104
Cumulative proportion	0.2285	0.3534	0.4358	...	0.9095

Table 1
Importance of the first few principal components.

separate the two classes. A few non-bents, along with a single bent, have PC_1 values above around fifteen, and are some distance away from the rest of the galaxies. Instead of being clustered together, however, these well-separated galaxies fill different regions of the PC_1-PC_2 space, and seem to be outliers. The other galaxies are mixed together in the plots, regardless of their class labels. On closer examination, the outlier galaxies have the largest values of the **peakFlux** variable, and correspond to the galaxies with the unusually high values of **peakFlux** visible in Figure 4.

Following [19], we used the PCs to eliminate unimportant variables. Considering the eigenvector corresponding to the smallest eigenvalue of the covariance matrix, we discarded the variable with the largest coefficient (in absolute value) in that vector. Then, we considered the eigenvector corresponding to the second smallest eigenvalue, and discarded the variable with the largest (in absolute value) coefficient, among the variables not discarded earlier. Continuing this process, we found the 20 most important variables to be:

(1) **angleAB**
(2) **angleAC**
(3) **ACRelDist**
(4) **BEllipt**
(5) **ABComDist**
(6) **ABRelPFlux**
(7) **ABAnglGeom**
(8) **ACRelPFlux**
(9) **ACAnglGeom**
(10) **BCRelIFlux**
(11) **BCRelEllipt**
(12) **BCRelSNR**
(13) **AIntFlux**
(14) **AEllipt**
(15) **ASidelobe**
(16) **BTotArea**
(17) **BIntFlux**
(18) **BRMS**
(19) **CIntFlux**
(20) **CMaj**

Comparing these features to the features suggested by EDA methods in Section 6.2.1, we note that the features **angleAB**, **angleAC**, **ACRelDist**, **AEllipt**, **BEllipt**, and **BRMS** are common to both methods.

6.3 Results with Classification Techniques

We next present the results for the classification of bent doubles, using the decision tree and GLM classifiers. We experimented with both the original set of features as well as the reduced sets found through EDA and PCA. In all cases, we performed 10-fold cross-validation 10 times. The errors we report include both misclassification errors: bents classified as non-bents, and non-bents classified as bents. The astronomers tolerate higher rates of the latter errors, but would like to minimize the mistakes of the former type.

6.3.1 Results Based on Decision Tree Classifiers

In our first set of experiments with decision trees, we used all but the book-keeping features. Results for a typical such tree is given below.

```
Decision tree created with all (but the book-keeping) features:

angleAC <= 4.656:
:...BEllipt <= 2.083: 1 (11.0)
:   BEllipt > 2.083: 5 (4.0)
angleAC > 4.656:
:...CPointSrc = 1:
    :...axialSym <= 3.038: 1 (5.0)
    :   axialSym > 3.038: 5 (3.0)
    CPointSrc = 0:
    :...BCRelFlux <= 3.512: 5 (145.0/4.0)
        BCRelFlux > 3.512:
        :...ACComDist <= 28: 5 (4.0/1.0)
            ACComDist > 28: 1 (4.0)
```

The output lists the feature selected at each node and the value it is compared against. The number after the colon indicates that the node is a leaf node, and the number is the class assigned to the leaf (5 denotes a bent double, while 1 denotes a non-bent double). At each leaf node, the numbers (a/b) indicate the (total number of samples/samples of the class not assigned to leaf node).

From 176 (=90% of the labeled data) cases in the training data, this particular tree correctly classified all bents, and all but 5 non-bents, with an overall error of 5/176=2.8%. This resubstitution error is overly optimistic, as it reflects the error on the same data that was used to build the tree (note that C5.0 uses pruning in constructing the trees). More relevant is the error on the 19 (=10% of the labeled data) test cases not included in constructing the tree. That cross-validation error is 1/19=5.3%, as the tree classified all bents from the test data correctly, and it misclassified 1 non-bent as a bent.

Results for ten different 10-fold cross-validation experiments are given under the "Tree 1: All features" columns in Table 2. The "Errors" column includes the two types of misclassification errors in the 10 ten-fold cross-validation experiments, and the "# Leaves" refers to the number of leaf nodes in the tree, as a measure of complexity of the trees. The Mean and the SE rows report the mean and the standard error, respectively, of the quantities in the corresponding columns.

The decision trees based on all features tend to pick combinations of angles and relative distances as the most important features. Other features include measures of ellipticity and symmetry — features that are all scale, rotation,

	Tree 1: All features		Tree 2: EDA, PCA features		Tree 3: GLM M3 features	
	# Leaves	Errors	# Leaves	Errors	# Leaves	Errors
Mean	7.8	11.1%	7.1	9.5%	6.7	8.3%
SE	0.1	0.4%	0.1	0.4%	0.1	0.4%

Table 2
Results of ten 10-fold cross-validation experiments using decision trees.

and translation invariant. The angles are usually either the core angle, or pairwise angles calculated geometrically — angles that are robust to small changes in the data. The very reason we included the geometrical angles, **AnglGeom**, is exactly to avoid the sensitivity of the differenced angles, **AnglDiff**, both explained in Section B.2. The trees generally ignore features related to the fluxes and the areas. Overall, the trees make sense, and they pick the features that we expected in the first place to be closely related to the problem.

To see the effect of reducing the number of features, we next created the tree using subsets of the features. First, we included the features selected by the EDA and PCA methods. The results are presented in the column "Tree 2: EDA, PCA features" in Table 2. A typical tree is given below. From 176 cases in the training data, this tree classified all bents correctly and incorrectly classified 3 non-bents as bents, with an overall error of 1.7% on the training data. From the 19 test cases, it classified all non-bents correctly, while misclassifying 1 bent as a non-bent, giving an overall error of 5.3% on the test set.

```
Decision tree created with the features suggested by EDA and PCA:

angleAC <= 4.656:
:...BEllipt <= 2.083: 1 (9.0)
:   BEllipt > 2.083: 5 (4.0)
angleAC > 4.656:
:...ACRelDist <= 6.427: 5 (143.0/3.0)
    ACRelDist > 6.427:
    :...BEllipt > 3.408: 5 (5.0)
        BEllipt <= 3.408:
        :...ABAnglGeom <= 120: 1 (10.0)
            ABAnglGeom > 120:
            :...BTotArea <= 22.84: 5 (3.0)
                BTotArea > 22.84: 1 (2.0)
```

Corresponding results, using only the features in the GLM model M3 in Section 6.3.2, are reported below, and in the "Tree 3: GLM M3 features" column in Table 2. From 176 cases in the training data, the tree below correctly classified all bents and incorrectly classified 5 non-bents as bents, with an overall

error of 2.8% on the test data. It correctly classified all the 19 test cases, resulting in a zero error on the test data.

```
Decision tree created with the features in the GLM model M3:

angleAC <= 3.999:
:...BEllipt <= 2.083: 1 (10.0)
:   BEllipt > 2.083: 5 (3.0)
angleAC > 3.999:
:...ACRelDist <= 6.401: 5 (141.0/3.0)
    ACRelDist > 6.401:
    :...BEllipt > 2.455: 5 (7.0/1.0)
        BEllipt <= 2.455:
        :...CRMS <= 0.136: 5 (3.0/1.0)
            CRMS > 0.136: 1 (12.0)
```

Comparing the results in Table 2, we find that including fewer variables in the model results in smaller and more accurate trees.

6.3.2 Results Based on Generalized Linear Models

For the FIRST dataset, we used the logistic GLM described in Section 5.2 to model the bent/non-bent categorical response variable as a function of the features. In our context, we were not as much interested in the goodness of fit of a model per se, as we were in finding a model with good predictive properties. In each fold of the cross-validation, we started by first fitting a fixed model, then used stepwise variable selection to find the best submodel. We used the *step()* function of the S-PLUS 6 software [26] that combines both backward and forward searches. This method starts with a user-specified model, then fits a series of other models by sequentially dropping (backward step) and adding (forward step) one variable at a time from a given list of variables. If the fitting improved from the previous model, the search is continued, otherwise the procedure stops. The goodness of a model is measured by an approximation to Akaike's Information Criterion (AIC), which, in principle, combines the negative of the log-likelihood with a penalty term for large models. A good model should minimize the AIC.

Before fitting a GLM to the FIRST features, we first used the pair-wise correlation matrix of the variables to combine the features suggested by EDA in Section 6.2.1 with the features suggested by PCA in Section 6.2.2 into a set of nearly uncorrelated features. We then formulated a GLM with the resulting features as the explanatory variables. This resulted in the model M1 described in Appendix C.

Next, we created model M2 from M1 by performing stepwise model selection.

	GLM M2	GLM M3	GLM M4
	Errors	**Errors**	**Errors**
Mean	0.947%	7.84%	4.00%
SE	0.22%	0.91%	1.14%

Table 3
Results of ten 10-fold cross-validation experiments using generalized linear models.

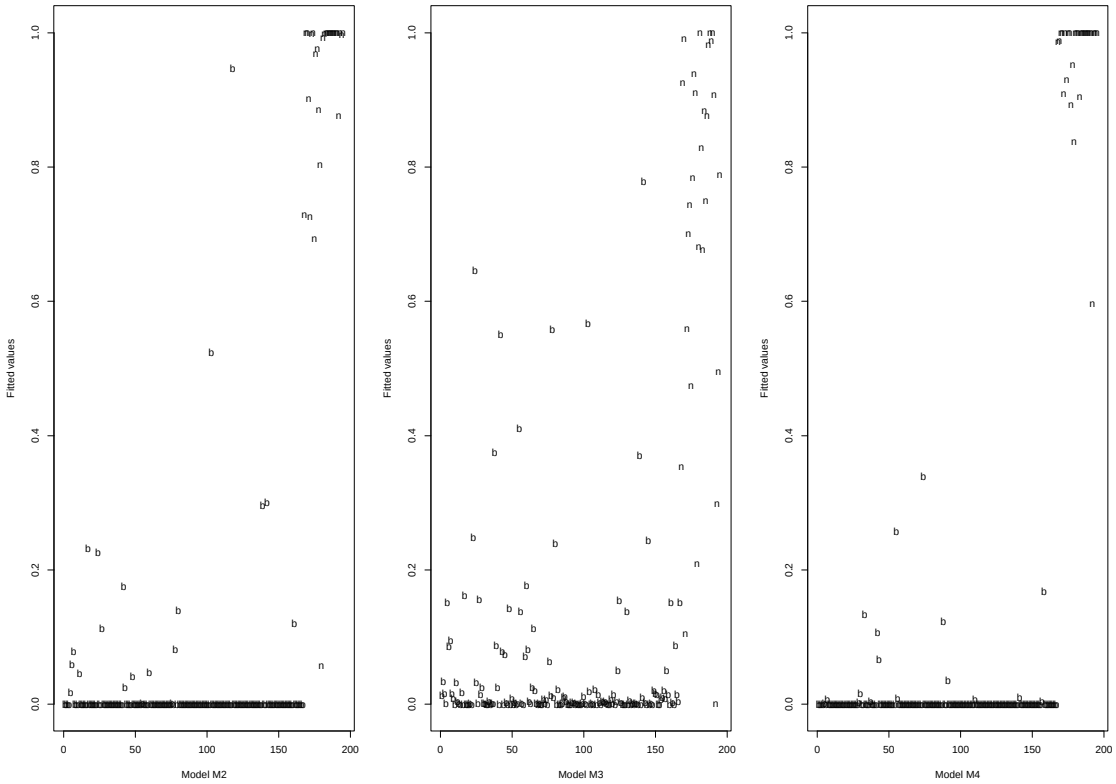


Fig. 7. Fitted values based on models M2, M3 and M4. In the S-PLUS parameterization, the class of bents is coded with zero, and the class of non-bents with one. The labels "b" and "n" indicate bents and non-bents, respectively.

As shown in Appendix C, M2 discards **ADiffusion** and **ASidelobe**.

The first column of Table 3 presents the misclassification errors of 10 ten-fold cross-validation experiments using M2. The resulting errors are much lower than the errors obtained using decision trees in Table 2, Section 6.3.1.

The first panel of Figure 7 presents the fitted values $\hat{\mu}_i$ in the interval $[0, 1]$, obtained via Eq. (12) from the estimated coefficients from M2. Note that in the S-PLUS parameterization, the class of bents is coded with zero, and the class of non-bents with one. Since we built the model M2 on the labeled training set,

we know the actual bent (b) or non-bent (n) class label corresponding to each galaxy. By comparing the true labels to the model-based fitted (predicted) values, we can study the accuracy of the model. Ideally, all bents should have fitted values close to zero, and all non-bents should have fitted values close to one. Using a cut-off value of 0.5, classifying the fitted values above 0.5 as non-bents, and the ones below it as bents, this model misclassified two bents and one non-bent.

In order to find alternative linear models, we also tried stepwise model selection starting from the five variables selected by both the EDA and the PCA feature selection methods in Sections 6.2.1 and 6.2.2, respectively. The best model we found with this procedure is M3 given in Appendix C. The corresponding fitted values are shown in the second panel of Figure 7.

Although most fitted values based on M3 corresponding to bents are close to zero, there are quite a few of them around 0.2, some around 0.6, and one close to 0.8. Most fitted values for the non-bents are clustered from 0.6 to 1.0, but some are below 0.6, and one is actually very close to 0. Based on Figure 7, we could hypothesize that galaxies with fitted values in the $[0, 0.2]$ interval are very likely to be bent doubles, those in the $[0.8, 1.0]$ range are very likely to be non-bent doubles, while the ones falling in between these two intervals can be either. If we wanted to use the fitted values to assign a unique bent/non-bent label to all galaxies, we could use a cutoff value, and round all fitted values below/above the cutoff to 0/1. Recall that we want to catch all the bents without missing any, but do not mind if we include occasional non-bents. From the second panel of Figure 7, the cutoff choice of 0.8 seems reasonable for M3. If we had used a cutoff of 0.5, this model would have misclassified five bents and seven non-bents. The second column of Table 3 summarizes the results of 10 ten-fold cross-validation experiments using M3.

Stepwise model selection considering adding or dropping single terms in M3 does not lead to any improvement. However, including up to second-order interaction terms [25,20,26] in the stepwise search leads to superior models, such as M4 given in Appendix C.

The fitted values based on M4 are shown in the third panel of Figure 7. Note the improved accuracy over the results of M2 and M3 in the first and second panels. Using a cutoff of 0.5, M4 classifies correctly all the galaxies. To investigate its predictive accuracy, the third column in Table 3 summarizes the results of 10 ten-fold cross-validation experiments starting from M3, and considering up to second-order interaction terms in the stepwise model selection.

Based on the results in Table 3, the model M2 leads to the lowest misclassification errors, followed by M4, then by M3. Compared with the results in Table 2 based on decision trees, on the average, generalized linear models tend to have

	Tree 1	Tree 2	Tree 3	GLM M2	GLM M3	GLM M4	All 6
Non-bent	5412	4628	5660	5118	11080	4340	637
Bent	9647	10431	9399	9941	3979	10719	2577

Table 4
Classification results for the 2000 catalog by the six different methods.

smaller errors. However, the tree and the linear model constructed using the same M3 features have comparable performance: compare the errors and their standard errors in the third column in Table 2 with the corresponding values in the second column in Table 3.

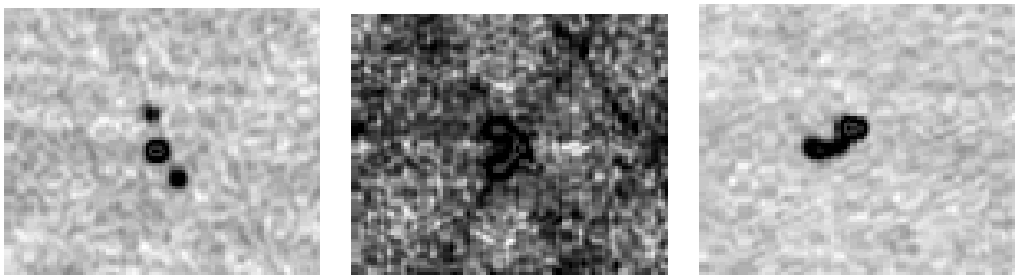
We also fitted GLMs using the 20 most important PCs (Section 6.2.2) as the explanatory variables. The average misclassification error of ten repetitions of such 10-fold cross-validation experiments was 11.63, with a standard error of 1.41, much higher than the errors reported in this section.

6.3.3 Classifying the Unlabeled Data

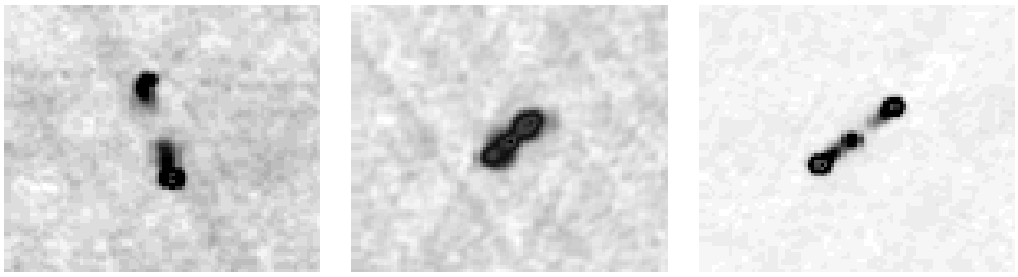
This section reports the classification results on the unlabeled 2000 catalog data. There are 15,059 three-entry radio sources in the catalog. Table 4 compares the number of galaxies classified by the six different classifiers: trees 1 through 3 are the decision trees in Table 2, and GLM M2, M3, and M4 are the generalized linear models in Table 3. The last column, labeled "All 6", refers to the cases where all six classifiers agreed. We have greater confidence in this last column, as six different classifiers, from two different categories, gave the same results. Figure 8 shows a few bent and non-bent galaxies similarly identified by all classifiers. In those cases, the class labels commonly assigned by the six methods are correct: the galaxies in the top row are indeed bents, while the ones in the bottom row are non-bents. To properly estimate the percentage of false positives and missed bents in the catalog, we would need to visually explore moderate-sized random samples from both the set classified as bents, and the one classified as non-bents.

7 Summary

In this paper, we described the role dimension reduction techniques can play in data mining using an application from astronomy, namely, the detection of radio galaxies with a bent double morphology. Our experiences indicate that the identification and extraction of relevant features plays a very important role in the accuracy of the pattern recognition algorithms. Equally important is the availability of a good training set, which can be non-trivial in scientific



(a) (b) (c)



(d) (e) (f)

Fig. 8. Example classification results for the unlabeled FIRST catalog data: (a)-(c) classified as bent doubles, (d)-(f) classified as non-bent doubles.

surveys.

Though much remains to be done, our initial results are very promising. We identified over 2500 potential "bent double" radio galaxies, a significant reduction from the more than 15000 galaxies in the survey. While we realize that there are inherent errors and uncertainties involved in every classification method, our work narrows the field considerably for the astronomers as they can now focus their observations on these galaxies to confirm their "bentness". In particular, we were pleased to notice that our semi-automated techniques identified a bent double missed in the manual search, thus illustrating the full potential of data mining. Our future plans include enhancing the training set through the examples validated by the astronomers, improving the feature set for the three-entry radio sources, and conducting a similar process for the two-entry radio sources.

Acknowledgments

We gratefully acknowledge our FIRST astronomer collaborators Robert Becker, Michael Gregg, David Helfand, Sally Laurent-Muehleisen, and Richard White for their technical interest and support of this work. Earlier versions of parts of this manuscript appeared in [9,14].

UCRL-JC-144209. This work was performed under the auspices of the U.S.

Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] Becker, R.H., R.L. White, and D.J. Helfand, "The FIRST Survey: Faint Images of the Radio Sky at Twenty-cm," *Astrophysical Journal*, Vol. 450, September, pp. 559-577, 1995.

[2] Bradley, P.S., O.L. Mangasarian, and W.N. Street, "Feature Selection via Mathematical Programming", *INFORMS Journal on Computing*, Vol. 10, pp. 209-217, 1998.

[3] Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone, "Classification and Regression Trees", CRC Press, Boca Raton, Florida, 1984.

[4] Burl M., L. Asker, P. Smyth, U. Fayyad, P. Perona, L. Crumpler, and J. Aubele, "Learning to Recognize Volcanoes on Venus", *Machine Learning*, Vol. 30, pp. 165-195, 1998.

[5] Chernoff, H., "The Use of Faces to Represent Points in K-dimensional Space Graphically", *Journal of the American Statistical Association*, Vol. 68, pp. 361-368, 1973.

[6] Dobson, A.J., "An Introduction to Generalized Linear Models", Chapman and Hall, 1990.

[7] Donoho, D. "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality", *Math Challenges of the 21st Century*, August 2000, http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/AMS2000.html.

[8] Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth, "The KDD Process for Extracting Useful Knowledge from Volumes of Data", *Communications of the ACM, Special Issue on Data Mining*, Vol. 39, pp. 27-34, 1996.

[9] Fodor, I.K., E. Cantú-Paz, C. Kamath, and N.A. Tang, "Finding Bent Double Radio Galaxies: A Case Study in Data Mining" to be published in *Computing Science and Statistics*, Vol. 33, 2000.

[10] Gahegan, M., M. Harrower, and T. Rhyne, "The Integration of Geographic Visualization with Databases, Data Mining, Knowledge Construction and Geocomputation", http://www.geog.psu.edu/~mark/ICA/ICAgroup993.htm.

[11] Hoaglin D.C., F. Moesteller, and J.W. Tukey, "Understanding Robust and Exploratory Data Analysis", *Wiley Series in Probability and Mathematical Statistics*, 1983.

[12] Inselberg, A. and B. Dimsdale, "Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry" *Proceedings of the First IEEE Conference on Visualization*, pp. 361-378, 1990.

[13] Jolliffe, I.T., "Principal Component Analysis", *Springer-Verlag*, 1986.

[14] Kamath, C., E. Cantú-Paz, I.K. Fodor, and N.A. Tang, "Searching for Bent Double Galaxies in the FIRST Survey", to be published in *Data Mining for Scientific and Engineering Applications*, Eds. R. Grossman, C. Kamath, W. Kegelmeier, V. Kumar and R. Namburu, Kluwer Academic Publishers, 2001.

[15] Kamath, C. and R. Musick, "Scalable Data Mining through Fine-Grained Parallelism: The Present and the Future," in *Advances in Distributed and Parallel Knowledge Discovery*, Eds. H. Kargupta and P. Chan, AAAI Press/MIT Press, pp. 29-77, 2000.

[16] Kohavi, R. and G. John, "The Wrapper Approach" in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Liu, H. and Motoda, H., Eds. Springer Verlag, 1998.

[17] Langley, P. and H. A. Simon, "Applications of Machine Learning and Rule Induction", *Communications of the ACM*, Vol. 38, No. 11, pp. 55-64, 1995.

[18] Lehar, J., A. Buchalter, R.G. McMahon, C.S. Kochanek, and T.W.B. Muxlow, "An Efficient Search for Gravitationally Lensed Radio Lobes", *The Astrophysical Journal*, Vol. 547, January, pp. 60-76, 2001.

[19] Mardia, K.V., J.T. Kent, and J.M. Bibby, "Multivariate Analysis", *Academic Press*, 1995.

[20] McCullagh, P. and J.A. Nelder, "Generalized Linear Models", *Chapman and Hall, London*, 2nd edition, 1989.

[21] Murthy, K. V. S., "On Growing Better Decision Trees from Data", Ph.D. thesis, Johns Hopkins University, 1997.

[22] Quinlan, J. R., "Induction of Decision Trees", *Machine Learning*, Vol. 1, pp. 81-106, 1986.

[23] Quinlan, J.R., "C4.5: Programs for Machine Learning", Morgan Kaufman, San Mateo, California, 1993.

[24] "Sapphire: Large Scale Data Mining and Pattern Recognition" website, *http://www.llnl.gov/casc/sapphire*.

[25] Seber, G.A.F., "Linear Regression Analysis", *Wiley Series in Probability and Mathematical Statistics*, 1977.

[26] S-PLUS 6.0 for UNIX User Manual, Data Analysis Division, MathSoft, Inc., Seattle, Washington, 2000.

[27] Tukey, J.W., "Exploratory Data Analysis", *Addison-Wesley*, 1977.

[28] White, R.L., R.H. Becker, D.J. Helfand, and M.D. Gregg, "A Catalog of 1.4 GHz Radio Sources from the FIRST Survey," *Astrophysical Journal*, Vol. 475, February, pp. 479-493, 1997.

[29] Witten, I.H. and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufman, 1999.

Appendix A: The Iteratively Re-weighted Least Squares Algorithm

In this section, we describe the IRLS algorithm to estimate the parameters of a GLM [20,6]. The notation and assumptions are as introduced in Section 5.2.

The parameters of a GLM can be estimated by maximizing the likelihood function corresponding to the assumed model. Let $d(y_i;\boldsymbol{\beta})$ denote the probability density function of the observation y_i given the parameter vector $\boldsymbol{\beta}$. The log-likelihood is then defined as

$$l(\boldsymbol{\beta}; y_i) = \log\, d(y_i;\boldsymbol{\beta}). \tag{14}$$

For n independent observations, the joint log-likelihood is

$$l(\boldsymbol{\beta}; \mathbf{y}) = \sum_{i=1}^{n} l(\boldsymbol{\beta}; y_i). \tag{15}$$

The parameter estimates $\hat{\boldsymbol{\beta}}$ maximize the joint log-likelihood, and are obtained by solving the normal equations:

$$\frac{\partial l(\boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\beta}} = \mathbf{0}. \tag{16}$$

It can be shown [6] that for a GLM

$$\frac{\partial l(\boldsymbol{\beta}; \mathbf{y})}{\partial \beta_j} = U_j = \sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{\mathrm{Var}(y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right), \quad \text{for } j = 0, \cdots, p-1. \tag{17}$$

The normal equations for the GLM are nonlinear in the parameters $\boldsymbol{\beta}$, and therefore the optimization problem has no closed-form solution. In practice, the parameters can estimated by iteratively re-weighted least squares (IRLS), a variant of the Newton algorithm.

The multidimensional version of the Newton method gives the mth approximation in terms of the $(m-1)$st approximation as

$$\hat{\boldsymbol{\beta}}^{(m)} = \hat{\boldsymbol{\beta}}^{(m-1)} - \left[\frac{\partial^2 l}{\partial \beta_j \partial \beta_k}\right]^{-1}_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{(m-1)}} \mathbf{U}^{(m-1)}, \tag{18}$$

where $\mathbf{U}^{(m-1)}$ is the vector of first derivatives U_j evaluated at $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}^{(m-1)}$. The method of scoring simplifies the Newton procedure by replacing the matrix of

second derivatives in Eq. (18) with its expected value,

$$\left[\frac{\partial^2 l}{\partial\beta_j\partial\beta_k}\right] \to \mathrm{E}\left[\frac{\partial^2 l}{\partial\beta_j\partial\beta_k}\right], \tag{19}$$

which by

$$\boldsymbol{\mathcal{I}}_{jk} = \mathrm{E}(U_j U_k) = \mathrm{E}\left[\frac{\partial l}{\partial\beta_j}\frac{\partial l}{\partial\beta_k}\right] = -\mathrm{E}\left[\frac{\partial^2 l}{\partial\beta_j\partial\beta_k}\right] \tag{20}$$

is the negative of the Fisher information matrix $\boldsymbol{\mathcal{I}} = \mathrm{E}(\mathbf{U}\mathbf{U}')$ of the U_js. The approximation in Eq. (18) therefore becomes

$$\hat{\boldsymbol{\beta}}^{(m)} = \hat{\boldsymbol{\beta}}^{(m-1)} + [\boldsymbol{\mathcal{I}}^{(m-1)}]^{-1}\mathbf{U}^{(m-1)}, \tag{21}$$

where $\boldsymbol{\mathcal{I}}^{(m-1)}$ is the information matrix evaluated at $\hat{\boldsymbol{\beta}}^{(m-1)}$. Pre-multiplying both sides by $\boldsymbol{\mathcal{I}}^{(m-1)}$, we obtain

$$\boldsymbol{\mathcal{I}}^{(m-1)}\hat{\boldsymbol{\beta}}^{(m)} = \boldsymbol{\mathcal{I}}^{(m-1)}\hat{\boldsymbol{\beta}}^{(m-1)} + \mathbf{U}^{(m-1)}. \tag{22}$$

Evaluating the expression of the information matrix for a GLM, we have

$$\boldsymbol{\mathcal{I}}_{jk} = \sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{\mathrm{Var}(y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2, \tag{23}$$

leading to

$$\boldsymbol{\mathcal{I}} = \mathbf{X}'\mathbf{W}\mathbf{X}, \tag{24}$$

where the diagonal $n \times n$ weight matrix $\mathbf{W}$ has elements

$$w_{ii} = \frac{1}{\mathrm{Var}(y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2. \tag{25}$$

By the results of Eq. (17) and (23), the elements of the right hand side of (22) are given by

$$\sum_{k=0}^{p-1}\sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{\mathrm{Var}(y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2 \hat{\beta}_k^{(m-1)} + \sum_{i=1}^{n} \frac{(y_i-\mu_i)x_{ij}}{\mathrm{Var}(y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right). \tag{26}$$

Writing

$$z_i = \sum_{k=0}^{p-1} x_{ik}\hat{\beta}_k^{(m-1)} + (y_i-\mu_i)\left(\frac{\partial\eta_i}{\partial\mu_i}\right) = g(\mu_i) + (y_i-\mu_i)\left(\frac{\partial g(\mu_i)}{\partial\mu_i}\right), \tag{27}$$

and substituting the results of Eq. (23) through (27) into Eq. (22), leads to the normal equations

$$\mathbf{X}'\mathbf{W}\mathbf{X}\hat{\boldsymbol{\beta}}^{(m)} = \mathbf{X}'\mathbf{W}\mathbf{z}, \tag{28}$$

where $\mathbf{z} = \{z_1, \ldots, z_n\}'$.

Comparing Eq. (28) to Eq. (6), note that the terms in Eq. (28) correspond to taking the diagonal covariance matrix $\mathbf{W}^{-1}$ as $\mathbf{\Sigma}$, and the transformed variable $\mathbf{z}$ as the response $\mathbf{y}$ in Eq. (6). In general, both $\mathbf{z}$ and $\mathbf{W}$ depend on $\hat{\boldsymbol{\beta}}$, and the solution is found by iteration. The steps for the logistic regression are as follows:

(1) Set initial values
- ${\mu_i}^{(0)}$; to avoid zero denominators in fitting the $\{0, 1\}$ variables, ${\mu_i}^{(0)}$ is generally chosen to be either 0.5 for all values, or to have two different values for the two levels of the response (say ${\mu_i}^{(0)} = 0.1$ if $y_i = 0$ and ${\mu_i}^{(0)} = 0.9$ if $y_i = 1$)
- $z_i^{(0)} = g(\mu_i^{(0)}) = \log\left(\frac{\mu_i^{(0)}}{1-\mu_i^{(0)}}\right)$
- $\mathbf{W}^{(0)} = \mathrm{diag}(w_{11}^{(0)}, \cdots, w_{nn}^{(0)})$, with $w_{ii}^{(0)} = \mu_i^{(0)}(1 - \mu_i^{(0)})$

(2) Form coefficient estimate as $\hat{\boldsymbol{\beta}}^{(1)} = (\mathbf{X'W^{(0)}X})^{-1}\mathbf{X'W}^{(0)}\mathbf{z}^{(0)}$

(3) For $i = 1, \ldots, n$, update the estimates as
- $\mu_i^{(1)} = \exp\{\mathbf{x}_i\hat{\boldsymbol{\beta}}^{(1)}\}/[1 + \exp\{\mathbf{x}_i\hat{\boldsymbol{\beta}}^{(1)}\}]$
- $z_i^{(1)} = \log\frac{\mu_i^{(1)}}{(1-\mu_i^{(1)})} + (y_i - \mu_i^{(1)})\frac{1}{\mu_i^{(1)}(1-\mu_i^{(1)})}$
- $w_{ii}^{(1)} = \mu_i^{(1)}(1 - \mu_i^{(1)})$.

(4) Iterate steps (2) and (3) until the difference between successive estimates $\hat{\boldsymbol{\beta}}^{(m-1)}$ and $\hat{\boldsymbol{\beta}}^{(m)}$ is negligible within the desired accuracy.

Appendix B: Features for Radio-Emitting Galaxies

B.1 Features for a Single Catalog Entry

The following list enumerates potential features pertaining to a single catalog entry. To differentiate among features corresponding to different entries in radio sources with more than one catalog entry, we prefix the feature names in such cases. For example, in a two-entry source, **APeakFlux** and **BPeakFlux** denote the peak fluxes corresponding to entries A and B, respectively.

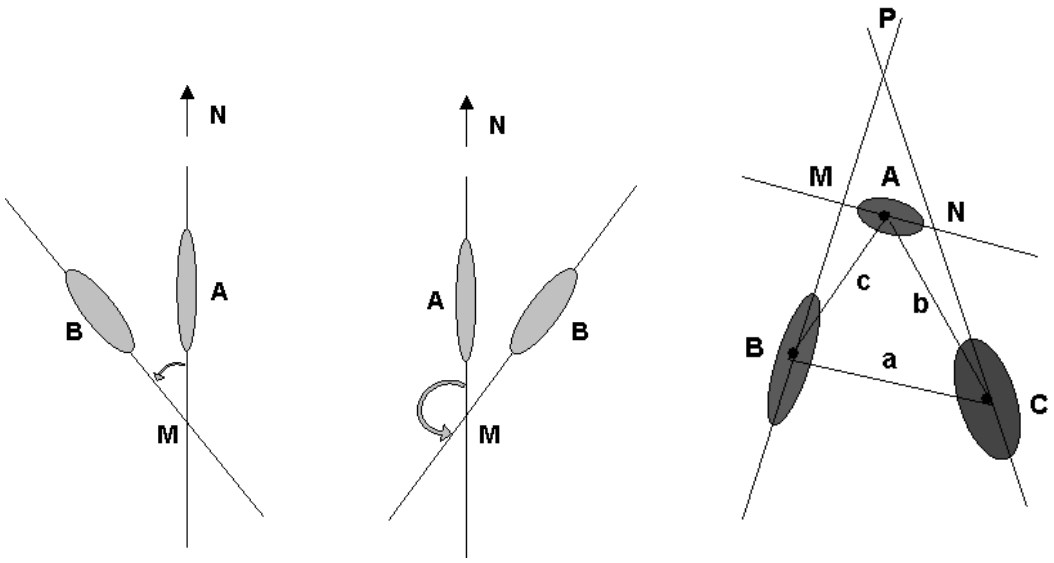
(1) **PeakFlux**: the peak flux value (mJy)
(2) **TotArea** $= \frac{\pi\ \mathbf{Maj}\ \mathbf{Min}}{4}$: the total area of the entry, as measured by the fitted elliptical Gaussian, where **Maj** and **Min** are the lengths of the major and of the minor axes, respectively
(3) **IntFlux**: the integrated flux value (mJy)
(4) **RA**: the right ascension RA (decimal hours)
(5) **Dec**: the declination Dec (decimal hours)

(6) **Ellipt** $= \frac{\mathbf{maj}}{\mathbf{min}} \geq 1$: a measure of the the entry's ellipticity, with one being a circular entry
(7) **RMS**: the local noise estimate (mJy) at the position of the entry in the sky
(8) **Sidelobe**: {0/1} flag, 1 if the entry might be a sidelobe of a nearby bright source, 0 otherwise
(9) **Maj**: the size (arc seconds) of the major axis
(10) **Min**: the size (arc seconds) of the minor axis
(11) **Diffusion** $= \frac{\mathbf{IntFlux}}{\mathbf{TotArea}}$: a measure of diffusion
(12) **SNR** $= \frac{\mathbf{PeakFlux} - 0.25}{\mathbf{RMS}}$: the peak flux density signal to noise ratio (the 0.25 reflects a bias correction documented in the FIRST survey); it can also be thought of as a "standardized" peak flux quantity
(13) **PointSrc**: {0/1} flag, 1 if the entry is a point source (its **Maj** less than 2 arc seconds), and 0 otherwise
(14) **Flux**: set to **PeakFlux** for point sources, and to **IntFlux** for elliptical sources
(15) **PosAngle**: the angle (degrees) of the major axis, measured counterclockwise from North — in Figure 9(a), the arrows indicate the angle corresponding to entry B: about 45^o in the left image, and about $(180 - 45)^o$ in the right one (0^o for entry A in both cases)

B.2 Features for Two Catalog Entries

Potential features for a 2-entry radio source, or for a 3-entry source with the entries considered two-at-a-time are listed below. Features (1) through (8) characterize a 2-entry radio source, and features (9) through (20) pertain to any two entries taken together. This distinction will become clearer in the 3-entry radio source case, Section B.3, where the meaning of features (1)-(8) below will change to include all three components, and, in addition, the radio source features will include all three combinations of the pairwise features (9)-(20). In a 3-entry source, to distinguish among the three sets of features (9) through (20), corresponding to the pairs AB, AC and BC, we prefix the feature names with the corresponding pair name. For example, the three **ComDist** features would be **ABComDist**, **ACComDist** and **BCComDist**. Figure 9(a) shows two possible elliptical Gaussian fits for a 2-entry source. The entries are ordered based on the maximum integrated flux, that is, entry A has higher integrated flux than entry B.

(1) **totArea**: the sum of the two total areas
(2) **peakFlux**: the max of the two peak fluxes
(3) **sumIntFlux**: the sum of the two integrated fluxes
(4) **avgDiffusion**: the mean of the two diffusions
(5) **totEllipt**: the sum of the two ellipticities



(a) (b)

Fig. 9. (a) Two examples of 2-entry fitted radio sources. (b) An example of a 3-entry fitted radio source.

(6) **maxFlux**: the max of the two fluxes
(7) **sideLobe**: {0/1} flag, 1 if at least one of the entries might be a sidelobe of a nearby bright source, 0 otherwise
(8) **pointSrc**: {0/1} flag, 1 if at least one of the entries is a point source (its **Maj** less than 2 arc seconds), and 0 otherwise
(9) **ComDist**: distance between the two centers
(10) **RelDist** $= \frac{4\ \mathbf{ComDist}}{\mathbf{AMaj+AMin+BMaj+BMin}}$: a measure of the relative distance between the two entries, values close to one indicating nearly intersecting entries
(11) **RelPFlux**: ratio of the two peak fluxes
(12) **RelFlux**: ratio of the two fluxes
(13) **RelMaj**: ratio of the two majors
(14) **RelIFlux**: ratio of the two integrated fluxes
(15) **RelEllip**: ratio of the two ellipticities
(16) **AnglGeom**: angle formed by the position angles of the two major axes, as calculated geometrically - angle AMB in both panels of Figure 9(a). For two point sources, we define this angle to be 180^o, and for a point source and a regular source to be 90^o.
(17) **AnglDiff**: angle formed by the position angles of the two major axes, as calculated by the absolute difference in the two position angles — about $|0-45|^o = 45^o$ in the left, and about $|0-135|^o = 135^o$ in the right panel of Figure 9(a)
(18) **AvgSNR**: the mean of the two signal to noise ratios
(19) **MaxSNR**: the largest of the two signal to noise ratios
(20) **RelSNR**: the ratio of the two signal to noise ratios

The features for 2-entry radio sources include the 20 pairwise features above and the 15 single entry features for each of the two components listed in Section B.1.

B.3 Features for Three Catalog Entries

There are different ways of ordering the entries in a 3-entry source. First, we need to identify the "core" of the galaxy. If we consider the triangle formed by the centers of the three Gaussians, the core is the entry opposite to the side that is the most unlike the other two sides in length. In the following, assume that A is the core. Figure 9(b), depicts a possible arrangement of the three catalog entries which is used to characterize the features reported below.

(1) **totArea**: the sum of the three total areas
(2) **peakFlux**: the max of the three peak fluxes
(3) **sumIntFlux**: the sum of the three integrated fluxes
(4) **avgDiffusion**: the mean of the three diffusions
(5) **totEllipt**: the sum of the three ellipticities
(6) **maxFlux**: the max of the three fluxes
(7) **sidelobe**: {0/1} flag, 1 if at least one of the entries might be a sidelobe of a nearby bright source, 0 otherwise
(8) **pointSrc**: {0/1} flag, 1 if at least one of the entries is a point source (its **Maj** less than 2 arc seconds), and 0 otherwise
(9) **coreAngl**: the core angle, defined as the angle BAC in the triangle above
(10) **angleAB**: angle ACB in the triangle above (between sides a and b)
(11) **angleAC**: angle ABC in the triangle above (between sides a and c)
(12) **totalBendGeom**: the total bentness of the source, equal to the sum of **ABAnglGeom** $= AMB$ and **ACAnglGeom** $= ANC$
(13) **totalBendDiff**: the total bentness of the source, equal to the sum of the pairwise angles **ABAnglDiff** $= |$**APosAngle** $-$ **BPosAngle**$|$ and **ACAnglDiff** $= |$**APosAngle** $-$ **CPosAngle**$|$
(14) **ariAngl** $= \text{acos}\frac{\text{BC}}{\text{AB+AC}}$: a measure of bentness suggested in [18]
(15) **ABAnglSide**: the angle formed by the major axis of B with the AB segment, angle ABM
(16) **ACAnglSide**: the angle formed by the major axis of C with the AC segment, angle ACN
(17) **sumComDist**: the sum of the three pairwise **ComDist**
(18) **sumRelDist**: the sum of the three pairwise **RelDist**
(19) **axialSym**: a symmetry measure given by the ratio of the ellipticities of entries B and C
(20) **ariSym** $= \frac{\text{AC}}{\text{AB}}$: a symmetry measure suggested in [18]
(21) **anotherSym** $= \frac{\text{AB+AC}}{\text{AB+BC+AC}}$: another symmetry measure

(22) **consDemote**: {0/1} flag, 1 if one of the non-core entries is far from the core, and 0 otherwise [B is considered far if AB $> 2 \times$ const $\times$ (**AMaj** + **BMaj**), where const is currently set to 3 arc seconds; similarly for C]

The features for 3-entry radio sources include the 22 triple feature above, the $3 \times$ [last 12, i.e. (9) $-$ (20)] pairwise features listed in Section B.2, and the 3×15 single ones in Section B.1. We therefore have a total of 103 features for each galaxy composed of three entries. Associated with each of those galaxies is a class label taking values in the set $\{5, 1, ?\}$, indicating whether the galaxy is a bent double $\{5\}$, a non-bent double $\{1\}$, or its status is unknown $\{?\}$.

Appendix C: GLM Fits Using the FIRST Features

The GLM model M1 obtained by fitting all the features obtained through EDA and PCA is given below. The rows indicate the $p = 28$ explanatory variables (including the intercept) of the model, the "Value" column reports the corresponding coefficient estimate $\hat{\beta}$, the "Std. Error" its associated standard error estimate, and the "t value" column the ratio of the first two columns. Under certain assumptions, those ratios follow a $\mathcal{T}$ distribution with $n - p$ degrees of freedom. The deviance is a measure of the goodness of fit based on the likelihood of the data. The null deviance is the largest deviance, corresponding to the intercept-only model. It is a reference measure, used to gauge the reduction in deviance achieved by the model with p parameters.

```
Coefficients for Model M1:
                    Value Std. Error    t value
  (Intercept)  -57.093412  37.928156 -1.5053042
    totEllipt   50.033907  14.847688  3.3698113
totalBendGeom   -7.150701   2.642374 -2.7061652
      ariAngl -317.925771  97.404147 -3.2639860
      angleAB  107.898636  34.284436  3.1471609
      angleAC  165.799156  52.442448  3.1615450
   ABAnglSide    9.500516   3.013814  3.1523233
   ACAnglSide   -7.664272   2.518944 -3.0426525
    ACRelDist   26.369560   7.426587  3.5506967
     axialSym   -8.451852   2.863872 -2.9511971
       ariSym   10.480015   3.133147  3.3448839
   anotherSym   83.863853  24.472044  3.4269248
     pointSrc    9.098698   3.348426  2.7173059
   ADiffusion    6.751144   5.195230  1.2994891
      BEllipt  -53.679285  14.998247 -3.5790372
         CRMS   20.544570   6.662908  3.0834238
    ABComDist   -4.508462   1.891256 -2.3838455
```

```
   ABRelPFlux   -6.777815   6.434170 -1.0534094
   ABAnglGeom   -1.896162   1.603219 -1.1827222
   ACRelPFlux  -65.771404  23.279409 -2.8253039
  BCRelEllipt   15.015719   4.328118  3.4693412
     BCRelSNR   93.244738  28.998183  3.2155372
      AEllipt  -23.523837   7.447329 -3.1586944
    ASidelobe  -17.340847  34.304004 -0.5055051
     BTotArea   11.656726   5.212098  2.2364747
     BIntFlux    5.795060   2.973766  1.9487277
     CIntFlux   31.930109  14.578911  2.1901573
         CMaj   -9.215941   4.914342 -1.8753152

    Null Deviance: 160.4564 on 194 degrees of freedom
Residual Deviance: 20.81799 on 167 degrees of freedom
      AIC: 76.82
```

Stepwise model selection starting from M1 discards the **ADiffusion** and the **ASidelobe** variables, and results in the model M2.

```
Coefficients for Model M2:
                   Value Std. Error   t value
  (Intercept)  -54.315607  14.514604 -3.742135
    totEllipt   57.380683  15.395853  3.727022
totalBendGeom  -10.038605   3.015320 -3.329201
      ariAngl -449.883492 121.416119 -3.705303
      angleAB  154.036632  41.838396  3.681705
      angleAC  232.857736  63.221864  3.683184
   ABAnglSide   10.133813   2.859259  3.544210
   ACAnglSide  -11.172152   3.317167 -3.367980
    ACRelDist   38.306010  10.217002  3.749242
     axialSym  -12.921232   3.588907 -3.600325
       ariSym   14.167198   3.904558  3.628374
   anotherSym  119.691329  32.132912  3.724883
     pointSrc   13.442112   3.457794  3.887481
      BEllipt  -66.849313  17.627782 -3.792270
         CRMS   17.078123   4.829860  3.535946
    ABComDist   -8.715963   2.577743 -3.381238
   ABRelPFlux  -10.989428   4.179021 -2.629666
   ABAnglGeom   -3.362615   1.579319 -2.129155
   ACRelPFlux  -79.717591  22.530221 -3.538252
  BCRelEllipt   20.536618   5.504795  3.730678
     BCRelSNR  119.950339  33.619660  3.567863
      AEllipt  -28.066927   7.742900 -3.624860
     BTotArea    9.159150   2.794514  3.277546
```

```
     BIntFlux    6.456872   2.661301  2.426209
     CIntFlux   44.317889  13.901259  3.188049
         CMaj   -4.340964   2.868163 -1.513500

    Null Deviance: 160.4564 on 194 degrees of freedom
Residual Deviance: 16.14147 on 169 degrees of freedom
      AIC: 68.14
```

The model M3 is created by first starting with the common variables identified by both the EDA and the PCA methods, then applying stepwise model selection.

```
Coefficients for Model M3:
                  Value Std. Error   t value
(Intercept)  -4.7669653  1.0224322 -4.66237
    ariAngl -36.2784958 11.7473094 -3.08823
    angleAB  12.8672428  4.7648430  2.70045
    angleAC  17.1145607  6.9184605  2.47375
 ABAnglSide   0.6282016  0.3846913  1.63300
  ACRelDist   1.7231843  0.4754129  3.62460
 anotherSym  11.5488358  3.0428690  3.79537
   pointSrc   1.0692337  0.5178718  2.06466
 ADiffusion   1.2334447  0.9166225  1.34564
    BEllipt  -2.0299085  0.7699809 -2.63631
       CRMS   0.8742627  0.5626143  1.55392

    Null Deviance: 160.4564 on 194 degrees of freedom
Residual Deviance: 57.64784 on 184 degrees of freedom
      AIC: 79.65
```

The model M4 is created from M3 by stepwise model selection, including second-order interaction terms (indicated by colons).

```
Coefficients for model M4:
                         Value Std. Error    t value
         (Intercept)  -39.985636  17.440329 -2.292711
             ariAngl -436.293828 176.649213 -2.469831
             angleAB  136.892303  62.219834  2.200139
             angleAC  240.866311 101.408343  2.375212
          ABAnglSide  -11.352798  20.399232 -0.556530
           ACRelDist   11.685406   5.499055  2.124984
          anotherSym  148.458562  58.490923  2.538147
            pointSrc   -1.230441   5.236727 -0.234963
          ADiffusion    8.685167   7.683723  1.130333
             BEllipt  -20.203233   8.728673 -2.314582
```

```
  angleAC:anotherSym  -18.090278  10.260183 -1.763153
     angleAB:BEllipt   14.335118   8.727912  1.642445
  angleAB:ABAnglSide  -14.997816  17.734393 -0.845691
  ACRelDist:pointSrc   21.325322   9.032502  2.360954
ABAnglSide:ACRelDist   10.697313   5.250487  2.037394

    Null Deviance: 160.4564 on 194 degrees of freedom
Residual Deviance: 5.189981 on 180 degrees of freedom
      AIC: 35.19
```