ELSEVIER

Computers in Biology and Medicine 36 (2006) 1104–1125

Computers in Biology and Medicine

www.intl.elsevierhealth.com/journals/cobm

Machine learning in bioinformatics: A brief survey and recommendations for practitioners

Harish Bhaskar*, David C. Hoyle, Sameer Singh

School of Engineering, Computer Science & Mathematics, University of Exeter, Exeter EX4 4QF, UK

Abstract

Machine learning is used in a large number of bioinformatics applications and studies. The application of machine learning techniques in other areas such as pattern recognition has resulted in accumulated experience as to correct and principled approaches for their use. The aim of this paper is to give an account of issues affecting the application of machine learning tools, focusing primarily on general aspects of feature and model parameter selection, rather than any single specific algorithm. These aspects are discussed in the context of published bioinformatics studies in leading journals over the last 5 years. We assess to what degree the experience gained by the pattern recognition research community pervades these bioinformatics studies. We finally discuss various critical issues relating to bioinformatic data sets and make a number of recommendations on the proper use of machine learning techniques for bioinformatics research based upon previously published research on machine learning.

1. Machine learning in bioinformatics

Machine learning techniques have found widespread application in bioinformatics [1]. The diverse range of rapidly expanding data produced by modern molecular biology has fuelled a need for accurate classification and prediction algorithms. The accuracy of classification algorithms can be affected by a large variety of factors, some of which may be considered to be generic to any machine learning algorithm and applicable to research in other application domains. It is these generic factors that have received attention from the pattern recognition and machine learning research community over a large number of

* Corresponding author.

E-mail addresses: h.bhaskar@exeter.ac.uk (H. Bhaskar), d.c.hoyle@exeter.ac.uk (D.C. Hoyle), s.singh@exeter.ac.uk (S. Singh).

0010-4825/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2005.09.002

years. By contrast, routine application of machine learning techniques to large scale molecular biology data sets is a relatively new occurrence. In some bioinformatics applications the novel nature of the data will require novel modifications of existing algorithms and procedures, and in some invention of entirely new techniques of analysis. However, other bioinformatics problems and data sets are amenable to a direct application of established pattern recognition and machine learning procedures for which it is possible to make recommendations on standard procedures to be followed. Obviously such recommendations will be biased by their origins within the mainstream pattern recognition and machine learning research literature but we feel it is still valuable to, (a) make them, and, (b) ascertain to what extent standard procedures have been adopted when previously applied to bioinformatics data sets.

To address the latter point, in Section 2 we present the results from a survey of specific bioinformatics papers, with the aim of extracting a number of metrics relating to the recent machine learning research in bioinformatics. We have preferred to focus on cases in which machine learning techniques have been used in an empirical "black-box" fashion, rather than as a generative model of the data. The survey is based on research published in the 5 year period 1999–2004. A large proportion of such studies involve the use of gene expression data obtained using microarrays. Ultimately this is due to the popularity and impact of this new technology among experimental molecular biologists.

1.1. Microarray data in bioinformatics

Typically microarrays monitor simultaneously the expression levels of thousands of genes in an organism [2]. This is done by measuring the signal intensity of fluorescing molecules attached to DNA species (reverse-transcribed from extracted target mRNA or genomic DNA) that are bound to complementary strands of DNA localized to the surface of the microarray. Usually a ratio of intensities is calculated for each probe or gene, corresponding to two different labeled populations of reverse-transcribed mRNA. This is done as either two, single-colour hybridizations, or as a competitive two-colour hybridization. After capturing, or "scanning" of the spot intensities, the raw ratios or intensities undergo a round of pre-processing, termed "normalization", to remove systematic errors (bias) within the data [3,4]. Early application of microarrays to the study of human disease conditions rapidly revealed their potential as a medical diagnostic tool [5–7]. This is a class prediction problem to which supervised learning techniques are ideally suited—given a number of training data points corresponding to "healthy" and "disease" conditions and the associated gene expression measurements, can the class (healthy or disease) of a new patient be predicted solely on the basis of their gene expression levels? Feature selection is often applied to identify a limited subset of "markers" on which to base the diagnosis. Such feature selection is often intuitive but somewhat heuristic, e.g. identifying those probes or genes that are most significantly differentially expressed between the two classes, or contribute most significantly to the determination of a linear discriminant or principal component [5]. The learning problem is obviously extremely high-dimensional and training sets often limited in size. In such circumstances classifiers can generalize poorly yet in such medical applications there is a clear imperative for robust prediction methods. This has focused research on the use of large-margin classifiers such as support vector machines (SVMs) [8–10].

The publications examined in this survey all have a common theme of attempting to classify or cluster a data measurement in some fashion, given a number of inputs. Obviously error-free classification is not usually possible, but one aims at producing a classifier that generalizes well on unseen data. The large error rate of a classifier can be usually attributed to the inherent difficulty of the classification problem. However, in finite sample situations, the following factors can also degrade the performance of the classifier: (a)

small number of samples; (b) large number of features; (c) complexity of the classification rule (e.g. quadratic versus linear discriminant function); (d) presence of outliers and (e) inappropriate width for a classifier involving non-parametric kernel density estimation. In Section 3, we discuss a number of observations resulting from the survey of papers and comment on their connection to the factors listed above. In Section 3, we also give a number of general recommendations regarding the use of machine learning techniques. In Section 4, we conclude and summarize our findings and recommendations.

2. Survey of bioinformatics studies (1999–2004)

It is important to put the examinations of this study in the context of past work in the area of bioinformatics. We surveyed manuscripts from the leading journals Nature, Proceedings of the National Academy of Sciences USA, and Bioinformatics between January 1999 and December 2004. Manuscripts from Bioinformatics and Proceedings of the National Academy of Sciences USA journals were surveyed chronologically from the archive collection available online. Papers from Nature were collected on the basis of a related search query on the topic (e.g. classification of gene expression data, etc.). It should also be noted that only those papers that were available for free download were considered in this survey. Manuscripts were selected according to two main criteria, (i) relevance of the topic to the domain of survey, i.e. machine learning in bioinformatics. The research reported in the manuscripts had to be concerned with the *application* of machine learning techniques to real data sets. Thus, purely theoretical analyses and research were excluded from this survey. We also required that development or discussion of machine learning aspects was a large focus of each surveyed manuscript. (ii) Availability of summary statistics concerning the data sets analyzed—for a number of paper characteristics of the data sets used may not have been entirely clear. Some example topics from papers in our survey include:

- Use of artificial neural networks (ANNs) for identification of sub-cellular structure from fluorescent microscopic images [11]; prediction of solvent accessibility of amino acid residues in proteins [12]; classification and diagnosis of cancers using gene expression data [7]; determination of polymorphisms in individual patients of the human TP53 tumour suppressor gene using Affymetrix GeneChip hybridization intensities [13].
- Use of SVMs for functional classification of genes using gene expression data [8]; classification of cancers using gene expression data [10,14]; classification of protein quaternary structure [15].
- k-nearest neighbour (kNN) classification of colon cancer and leukemia samples using gene expression data [16,17]; β-turns in proteins [18].
- Clustering of gene expression data for the classification of B-cell lympoma samples [6]; breast tumour samples [19,20]; prostate cancer samples [21]; colon cancer samples and leukemia samples [22].

For the surveyed manuscripts we evaluated the following: (a) the total number of studies in the area of machine learning in each of the years that met our criteria; (b) the topic of classification (gene expression data classification or others); (c) the type of data set used (public or private data set); (d) the ratio of the number of samples used to the final number of features used (e) the use of data dimensionality reduction; (f) the problem of missing data; (g) the use of cross-validation approaches; (h) the use of supervised or unsupervised techniques of classification; (i) the classification rate

obtained; and (j) compression ratio (ratio of the number of reduced features to the original number of features. A full list of the manuscripts selected for this survey and summary statistics is available from http://www.dcs.ex.ac.uk/dchoyle/SuppInfo/CBM/Survey.xls. The results are presented in Fig. 1, from which we can make the following important observations:

- *The total number of studies in the area of machine learning in each of the years* (Fig. 1(a)): We find that the number of studies in the area of machine learning for solving problems in bioinformatics that met our survey criteria rapidly increased from 1999–2003. As the research field has matured and research has shifted focus from algorithm development and testing to application of algorithms and discovery of biological knowledge there has been a decrease, in 2004, in the number of papers that primarily discuss the performance of a particular machine learning classification or prediction algorithm. In 2004, we found an additional 19 papers that also utilized machine learning tools in the analysis of real data sets, but where the primary focus was discovery of biological knowledge through the application of machine learning rather than development of the machine learning tools used. Consequently we did not include these papers in our survey.
- *Topic of classification* (Fig. 1(b)): There is substantial amount of research in gene expression data analysis which is far greater in amount than protein data analysis and other topics. We suspect that is due to the wide public availability of gene expression data sets on which to test machine learning algorithms, as well as the increasing popularity of the microarrays as an assay for molecular biologists.
- *The type of data set used* (Fig. 1(c)): We define the public data set as that available for download on the Internet, whereas private data sets are personal to the authors and not available for purchase or download. It is important to point out that all public data sets are not necessarily benchmarks. It appears that authors frequently use both types of data sets (obviously more likely public data since it is easier to acquire). Private data sets are also being increasingly used (it is worrying that no further research or comparisons can be performed with such data).
- *The distribution of the ratio of number of samples to the number of reduced features* (Fig. 1(d)): The mean value of 2.2 (standard deviation 8.08) shows that the number of samples were often only twice the number of final set of features used. This is a very low figure—ratios of 10 or over were only found in a few studies.
- *Classification rate* (Fig. 1(e)): We find that almost all studies reviewed had classification rates of more than 80%. It seems that there is undoubtedly some pressure on authors to report only high recognition rates. In addition, the number of studies that have greater than 90% classification rates is more than those between 80–90% mark. We found hardly any studies that reported poor results.
- *Compression ratio—the ratio of final to original number of features* (Fig. 1(f)): Quite often it is difficult to know how many features to reduce a data set to. Ideally, one needs to follow a systematic procedure whereby a final subset of features is found which is optimal (the addition of further features does not increase the performance of the system). We did not find enough evidence that the studies reviewed followed a systematic approach for this purpose. Most of the studies seemed to substantially reduce the data set, to less than 10% of the original size. However several other studies did not reduce data enough and lacked a systematic methodology for determining feature redundancy.
- *The use of data dimensionality reduction* (Fig. 1(g)): It appears that several authors familiar with machine learning techniques did use data dimensionality reduction. However, we found that most studies used only principal components analysis (PCA) and were not particularly aware of other methods, even though other dimensionality reduction methods such as independent component analysis [23],

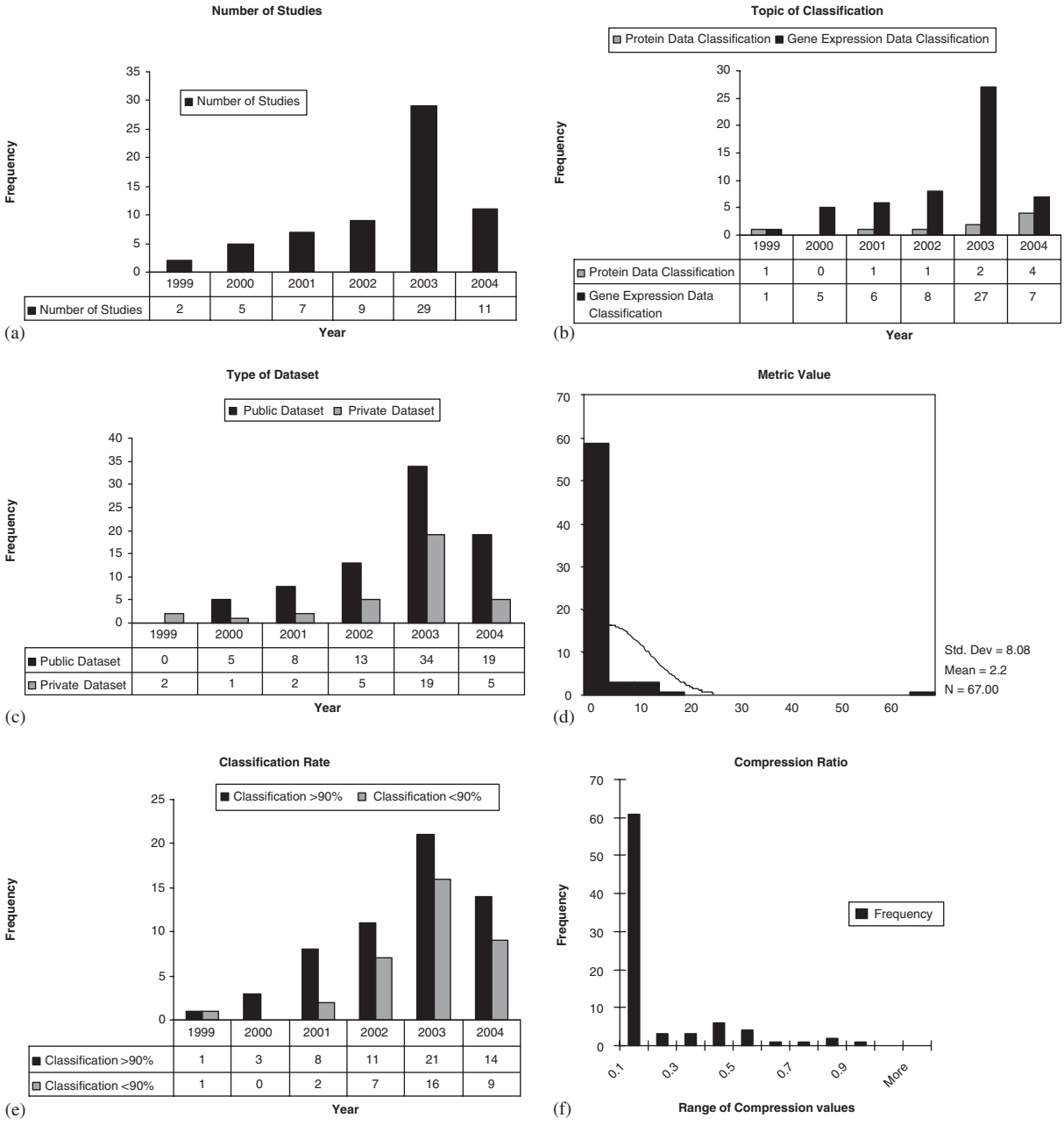
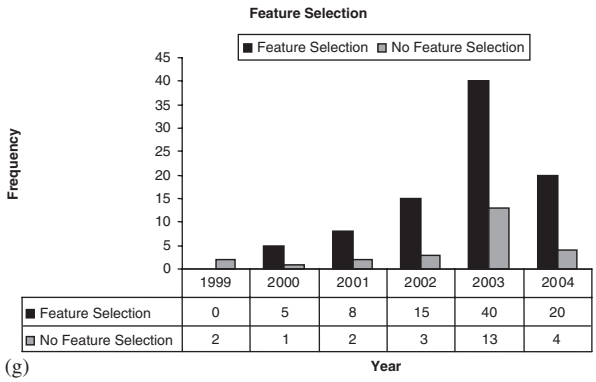
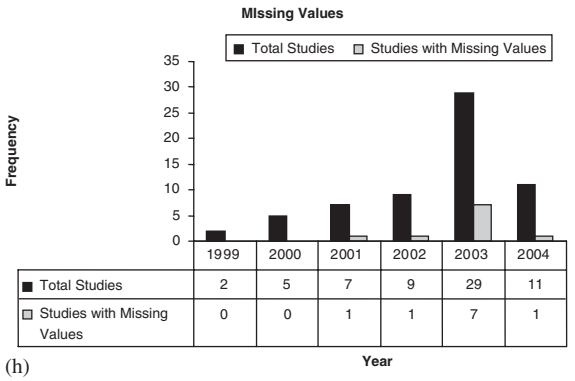


Fig. 1. Statistical analysis of published studies in the journals Nature, Proceedings of the National Academy of Sciences, USA, and Bioinformatics (1999–2004). Details of the content of each graph are given above.

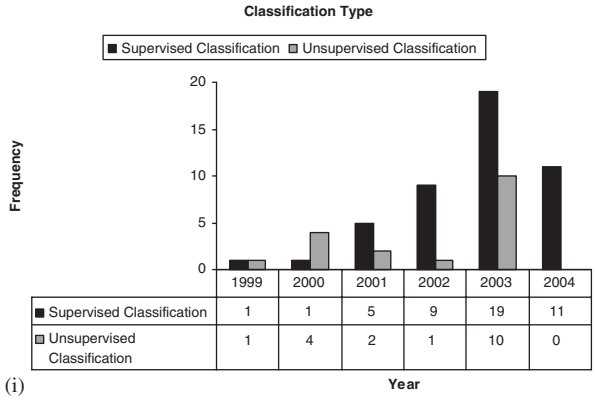
Sammon mapping [24], and self-organizing maps [25] have been reported in the bioinformatics literature. We also found that several studies with high data dimensionality and low number of samples did not reduce features at all.



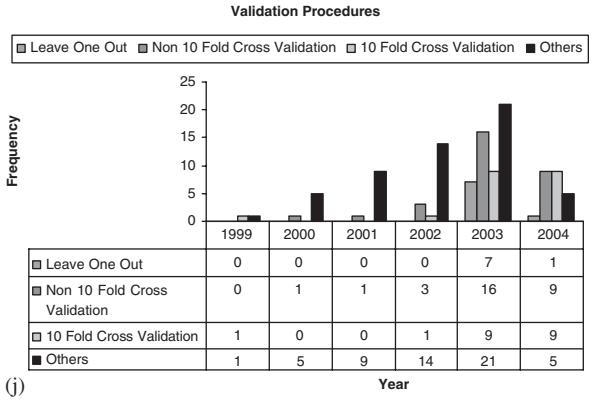
	1999	2000	2001	2002	2003	2004
Feature Selection	0	5	8	15	40	20
No Feature Selection	2	1	2	3	13	4



	1999	2000	2001	2002	2003	2004
Total Studies	2	5	7	9	29	11
Studies with Missing Values	0	0	1	1	7	1



	1999	2000	2001	2002	2003	2004
Supervised Classification	1	1	5	9	19	11
Unsupervised Classification	1	4	2	1	10	0



	1999	2000	2001	2002	2003	2004
Leave One Out	0	0	0	0	7	1
Non 10 Fold Cross Validation	0	1	1	3	16	9
10 Fold Cross Validation	1	0	0	1	9	9
Others	1	5	9	14	21	5

Fig. 1. (*continued*)

- *Missing values* (Fig. 1(h)): A significant number of studies have to deal with missing data. Typically this was for gene expression data sets. In such instances missing values may be replaced by imputed values
- *Machine learning approach* (Fig. 1(i)): In general, the number of studies using both supervised and unsupervised classification has increased. However, supervised classification is more common in use, particularly for analyzing gene expression data.
- *Use of cross-validation approaches* (Fig. 1(j)): There seems to be considerable inconsistency in the manner in which data results are analyzed. There is a general consensus that some form of validation is necessary, e.g. a good result could be merely based on chance, goodness of training data, or simplicity of test data. Hence, a number of trials with different training and test sets are necessary, and their results can be averaged. Studies have used different forms of cross-validation such as leave-one-out or 10-fold cross validation (in several cases this is an alternative to the leave-one-out method that is infeasible with classifiers such as neural networks), bootstrapping, and their own defined criteria. Unfortunately the choice of folds used, or the experimental set-up of bootstrapping is not always based on statistical principles.

3. Analysis of survey findings

Having extracted a number of measures from the survey of papers it is informative to see what, if any, general trends or striking features are apparent. Figs. 1(a)–(c) confirm the growth of machine learning in bioinformatics, and the general nature of problem solving and data sets used. Below we comment on a number of interesting aspects resulting from the survey and try to assess their impact upon the use of machine learning techniques in bioinformatics. We also give a number of recommendations based upon these observations and past experience drawn from the machine learning and pattern recognition community in general. To aid this process of recommendation we have summarized in Table 1 (at the end of the paper) the strengths, weaknesses and general properties of various machine learning classification and prediction algorithms, many of which were employed in the surveyed papers. In Table 1, we include machine learning tools for classification, clustering and dimensionality reduction. An exhaustive empirical comparison across all tools is beyond the scope of this paper and the results in such a case would be difficult to generalise because of data dependence rather than the ability of the tool. In Table 1, we present a qualitative comparison of the various tools based on the consensus in the opinion of four human experts in machine learning and our literature review. The tools are compared on the basis of (a) whether the parameters that need to be user set are small or large; (b) the boundary generated by tool to classify or reduce features is linear or non-linear; (c) whether the tool is unreliable in its analysis of data that has small sample to feature ratio and to what extent; (d) the time taken by the tool to learn from training data and test new samples; (e) whether the tool has an underlying data-based assumption for it to work well; (f) the extent to which outliers make the working of the tool unreliable; (g) whether the tool is mathematically well-defined and its learning can be easily understood or not; and (h) how easy is it to improve the tool with new data by performing partial and incremental learning rather than complete relearning. We have used fuzzy linguistic variables of "High", "Medium" and "Low", as well as "Good", "Medium" and "Poor"—the latter categories define qualitative concepts whereas the previous ones define quantitative concepts. These linguistic variables are relative in nature, and typical of the category they define, e.g. clustering or classification. Even though these categories are not precise, machine learning practitioners very well understand what they mean, and will find Table 1 as a good starting point to compare the relative advantage of different tools on a generic basis.

3.1. *Effects of sample size*

From Fig. 1(d), we can see that in the majority of publications surveyed the ratio of the number of samples to the number of features is less than ideal, even in many cases after feature selection has been performed. A limited number of training samples will ultimately impact upon the generalization ability and accuracy of any machine learning system.

3.1.1. *Critical issues and recommendations*

The ratio of the number of samples to the number of features should be fairly large for developing adequate machine learning systems. With a large number of features, one requires a complex system of data analysis, and the data-fitting problem (samples to classes) becomes tedious. The gene expression data sets surveyed typically consist of 10–100 sample points, with $O(10^4)$ potential explanatory variables. This is due to whole genome or nearly whole genome microarrays being available even for complex eukaryotes such as humans. In contrast, performing more than a handful of hybridizations in any experiment may

Table 1
Properties of machine learning classification and prediction algorithms

Classification tools	Parameters	Linear (L)/ non-linear (NL)	Effect of small sample/feature ratio	Computational complexity	Data assumptions	Noise and outlier effect	Transparency	Incremental learning
MLP neural network	High	NL	Medium	High	None	Low	Poor	Poor
RBF neural network	High	NL	Medium	Medium	None	Low	Good	Poor
Self-organising maps	Medium	NL		Medium	None	Low	Poor	Poor
Probabilistic neural networks	High	NL	Medium	Medium	None	Low	Good	Poor
Support vector machines	Low	L/NL	Low	Medium	Variable	Low	Good	Medium
Linear discriminant analysis	Low	L	Low	Low	Gaussian, equal variance	Medium	Good	Medium
Quadratic discriminant analysis	Low	NL	Low	Low	Gaussian unequal variance	Medium	Good	Medium
k nearest-neighbour	Low	NL	Low	High	None	Low	Good	Good
Gaussian mixture model	Medium	NL	High	High	Variable	High	Good	Poor
Naı¨ve bayes	Low	NL	High	Low	None	Low	Good	Poor
Decision trees	Low	NL	Medium	Medium	None	Low	Good	Poor
Neuro-fuzzy systems	Low	NL	Medium	High	None	Low	Good	Poor
Clustering tools								
Self-organising maps	Low	–	Medium	High	None	Low	Poor	Medium
k-means	Low	–	High	Medium	Spherical clusters	High	Poor	Good
Fuzzy *c*-means	Low	–	High	Medium	Spherical clusters	High	Poor	Good
Hierarchical clustering	Medium	–	High	Low	None	Low	Good	Good
Dimensionality reduction								
Principal component analysis	None	L	Low	Low	Gaussian densities	High	Good	Medium
Linear discriminant analysis	Low	L	Low	Low	Gaussian densities	High	Good	Poor
Sammon's mapping	Low	NL	Low	High	None	Medium	Poor	Poor
Multi-dimensional scaling	Low	NL	Low	Low	None	High	Good	Medium
Independent components analysis	Low	NL	Medium	Medium	Variable	High	Good	Medium

be prohibitive, either due to cost, time constraints, or the complexity and difficulty in obtaining the extracted RNA. However gene expression levels across the genome can be regulated by a much smaller number of factors, e.g. regulatory elements, with the consequence that the data has some intrinsically low-dimensional representation and unsupervised techniques such as principal component analysis (PCA) perform well on class prediction [26]. In general though the excess number of features compared to samples has major implications for the quality of supervised or unsupervised data analysis, and reliability of results. In addition, less statistical confidence can be placed in a study that uses a small number of data samples. Occasionally, for some classification problems one may be able to construct more appropriate features, such as BLAST or other alignment scores with prototype sequences. This can lead to improved representations and classification performance [27], but often in terms of data pre-processing there is only a limited number of things that can be done. Obviously, one can reduce the number of features, by selecting informative features from the original set, and try to develop a simpler system that does not overfit.

A number of studies have investigated the role of sample size in improving classification accuracy [28–30]. Several recommendations for practitioners are presented by Raudys and Jain [31]. These can be summarised as follows:

- Specific recommendations can be offered for specific classifiers as follows: (a) A classifier such as Euclidean distance classifier should be used when the classes are well-separated and a simple decision rule is needed; (b) Fisher's linear discriminant is asymptotically optimal for the classification of Gaussian populations with a common covariance matrix, however, the quadratic discriminant function suffers significantly from the non-normality of data. It is important to note that as the number of samples from different classes becomes imbalanced, the performance of the technique is further degraded; (c) the performance of a Parzen window classifier depends on the value of the window function and the smoothing parameter. Windows functions such as exponential and logistic have been frequently used. Generally the smoothing parameter is much more important than the window type—it should normally decrease as the number of samples used increases; (d) for the kNN classifier, the Mahalanobis metric often gives much better results compared to the Euclidean distance metric. For small sample sizes, the value of k is an important parameter and must be optimized; and (e) there is less amount of guidance available on the design of multinomial classifiers (e.g. where the data is categorized in histogram bins and matched). The classification error depends on the number of bins and the thresholds used.
- For a real problem, the estimate of sample size needed to solve a problem depends on the method used to find the parameters of the classification rule, the number of features, the asymptotic probability of misclassification (error rate), and the desired learning accuracy. Raudys and Jain [31] provide extensive discussion on the relationship between the sample size N and classification accuracy for a two class, Gaussian distribution data set. They show that the increase in classification error of the parametric classifiers is proportional to $1/N$ and depends on the dimensionality of the feature space p; for the linear classifiers, the relationship is linear and for quadratic classifiers the relationship is quadratic (only for large p). For non-parametric classifiers such as Parzen windows, k nearest-neighbour, the increase in classification error is proportional to $1/\sqrt{N}$ or $1/N^{\cdot 33}$. These estimates are for Gaussian distributions with equal covariances—additional factors will influence these estimates for data with unequal covariances and different number of samples per class. For other data distributions, and more than two class problems, it is recommended to estimate the following metric: $\hat{\Delta}_N = \hat{P}_c - \hat{P}_R/2$, where $\hat{\Delta}N$ is the increase in classification error, $\hat{P}_c$ is the leave-one-out estimate of classification error, and

$\hat{P}_R$ is the re-substitution estimate of the classification error. If the difference $\hat{\Delta}N$ is small in comparison with the empirical estimate of asymptotic probability of misclassification (error rate), then the sample size is sufficient.

In the context of microarray data analysis there are a number of systematic studies comparing the performance of various classifiers on gene expression data [32,33], to which we refer the reader for additional recommendations. In addition two important studies, Hwang et al. [34] and Mukherjee et al. [35], have investigated the minimum sample size needed for statistically reasonable data analysis. Hwang et al. [34] estimate the minimum number of array samples need to ensure satisfactory separation of disease subtypes when using linear discriminant analysis (LDA) for classification. The authors use power analysis [36–38] for the determination of minimum sample size (see Hwang et al. [34] for further details).

Mukherjee et al. [35] address a similar question: "What is the relationship between sample size and classification performance?" Approaches within the statistics and pattern recognition communities have used power calculations, for example the work by Adcock [39] and Guyon et al. [40], but assume data normality and independence of variables—assumptions that may not necessarily hold. They compute bounds or estimates of a quantity's deviation from its expected value as a function of the number of samples. Unfortunately, these methods are not suitable for predicting the future performance of a classifier as the sample size is increased. Therefore Mukherjee et al. [35] suggest the use of learning curves proposed by Cortes et al. [41]. Learning curves estimate the empirical error rate as a function of the training set for a given classifier and data set. These learning curves are well characterized by inverse-power laws: $e(N) = aN^{-\alpha} + b$, where $e(N)$ is the expected error rate, N is the number of samples, α is the learning rate, and b is the Bayes error which is the minimum error rate achievable. These parameters take on different values depending on the type of classifier and data set being used. As the data set increases in size asymptotically, the error rate approaches b. This equation holds well for a number of classifiers [41]. Using this power-law scaling model as a basis, one can use the empirical error rates of a classifier over a range of training set sizes drawn from a data set to fit an inverse-power law model. The fitted inverse-power law model can be used to extrapolate the error rate to larger data sets. Mukherjee et al. [35] develop the tests to detect when this model fails (especially with very small sample sizes), such that this part of the curve is ignored when fitting and extrapolating.

3.2. Feature selection and extraction

Fig. 1(g) shows that a large proportion of studies (nearly 25% in 2003) did not employ any feature selection procedure at all. It has been empirically shown in several studies that redundant or non-discriminatory features reduce the ability of classifiers to learn decision boundaries between data of different classes.

3.2.1. Critical issues and recommendations

Identification of relevant features is extremely important for classification tasks (improving accuracy and reducing computational costs), as well as for understanding the relative significance of features. For this purpose, feature selection has been often distinguished in the research literature from feature extraction. Feature selection is aimed at finding individually from a large group of features the best features that maximize classification ability. Feature extraction has been traditionally viewed as a process to use different weighting schemes to linearly (as in PCA and LDA) or non-linearly (as in neural networks) combine features to produce a reduced number of new, ideally uncorrelated, features.

The first critical issue then is whether one should use feature selection or feature extraction. There is no clear cut evidence that one of them is superior to the other on all types of tasks. They also have their own limitations. Feature extraction methods such as PCA can fail if the variability in some of the features is nearly zero, and data assumptions are not met. Similarly, for feature selection, exhaustive search, and most sequential search methods on high dimensional data sets are infeasible. Methods that examine features individually are simple to use, but yield poor feature subsets. On the other hand, those methods that try different feature combinations simply take too much time and run out of computer memory. Finally, features that appear to be important on training data may not fare that well on test data. We discuss below the different methods involved in dimensionality reduction and how best to apply the well-established techniques.

Feature selection: Feature selection schemes are often divided in two categories: (i) Filter methods which use only the general characteristics of the training data without reference to the learning algorithm that is finally used, (ii) Wrapper methods that make use of the performance of the chosen classifier to select features. Within the pattern recognition literature the most popular schemes for feature selection are based on *classification complexity estimation*, and *validation set classification*. In addition Markov Blanket filter methods have proved popular for use with gene expression data [42,43]. The methods based on the use of classification complexity work only with the training data for selecting the best features and so are filter methods. The aim of all such methods is to determine the averaged pair-wise of separation between different class distributions. These methods use two important criteria: a classification complexity metric, and a methodology for selecting feature subsets. A feature selection scheme usually iterates by choosing several subsets of features and estimating the classification complexity of the subset data. The aim is to determine the best subset in terms of the least classification complexity. The validation set classification methods use a validation set in addition to the training and testing data, and use a fixed classifier, and are therefore wrapper methods. For each subset of features considered, the classification accuracy using training data to train the classifier and validation data to test the classifier, is used as a metric of the quality of the feature subset.

Validation set based feature selection: In these schemes, a validation set is used for testing a system that has been trained on the subset of features that are being considered. A classifier with fixed parameters is used to obtain classification results. Within a bioinformatics context Xiong et al. [44] have evaluated a number of classification algorithms for feature selection, using the colon gene expression data of Alon et al. [45]. The accuracy can be improved by resampling techniques and the use of cross-validation. This method has however some difficulties. Firstly, obtaining a validation set that will truly relate to the real test data is quite hard. In other words, good results on the validation set do not necessarily promise good results on the test data. Secondly, it has been shown by Jain and Zongker [46] that the classifier parameters, without optimisation, will generate incorrect subsets of features. This can be a very time consuming tasks for all possible feature subsets considered. Finally, feature subsets selected using a given classifier may not work well with another classifier. Furthermore, the classifier may be biased depending on the amount and dimensionality of the data set. Hence, the preferred option is to use classification complexity estimates, such as multi-resolution approaches, that do correlate well with classification accuracy of classifiers, but are computed independent of them [47,48].

Feature subset selection methods: The simplest method of selecting features is to consider them in isolation. This technique is also commonly known as the best N technique. Each feature is individually tested and ranked in terms of its ability to separate data of different classes. At the other extreme, all possible subsets can be evaluated using an exhaustive search. In this case, for a total of d variables, and

selection of subsets of size p, the total number of possible subsets is equal to: $d!/(p-d)!d!$. Obviously, the number of subsets to be evaluated based on testing all subsets exhaustively is an enormous task. As such, a number of strategies have been proposed in the literature for selecting the best combination of features without testing all subsets. Jain and Zongker [46] provide a taxonomy of feature selection algorithms that distinguishes at the root level between methods based on neural networks and those based on statistical pattern recognition. Neural network approaches are based on pruning methods where successively each feature can be removed from analysis and its contribution to the overall classification accuracy can be judged [49]. Another simple approach uses the learnt weights to separate strong from weak features. In the statistical pattern recognition domain, exhaustive search and branch and bound methods produce optimal solutions [50]. The methods that give sub-optimal solutions either generate a single solution or multiple solutions. Single solutions generated in a deterministic manner are based on sequential searches with different feature subsets. A number of methods on sequential forward and backward searches have been explored and in their study on comparing several algorithms, Jain and Zongker [46] concluded that sequential forward floating search (SFFS) outperformed all other tested methods. Indeed Xiong et al. [44] utilized SFSS in their evaluation of wrapped feature selection for gene expression data. Obviously, the success of methods also depends on the metric used and it would be reasonable to suggest that choosing a good classification complexity metric is important for achieving good quality features, and choosing an appropriate feature subset search algorithm is important for ensuring high speed with which the solution is produced. Single solutions can also be generated in a stochastic manner using simulated annealing. Multiple solutions are generated in a deterministic manner using best first and beam searches. Genetic algorithms are used to generate stochastic output giving multiple solutions to the problem. A full review of the above techniques is available in Jain and Zongker [46].

Feature extraction: Another option for reducing data dimensionality is to simply use all of the available features and then combine them linearly to generate, ideally uncorrelated, set of features as with PCA [51]. In the last few years non-linear kernel PCA has also been extensively used [52]. PCA is an effective method for reducing data dimensionality and ensuring that the resultant features are uncorrelated. It is based on sound principles, and can be used as an effective data visualization technique. However, it makes assumptions on the nature of data and does not work very well when the covariance matrices cannot be reliably estimated due to small amounts of data, though robust variants exist and have been applied to gene expression data [53]. In addition, the principal components are not ideally suited for classification tasks, and can be error prone with noisy data. Non-linear PCA does solve some of these problems but represents a more tedious procedure for estimating principal components (optimal kernel function parameters may need to be estimated). Sammon mapping is another method of reducing data dimensionality by topologically retaining the structure of data in low dimensions [54,55,24]. The idea is to keep the relative distance between data points in higher dimension the same as in lower dimension. This process is however considerably slow and highly computationally expensive. In addition, as the ratio of the reduced dimensionality to original dimensionality decreases, the method generates very high errors. Finally, schemes such as linear discriminant analysis can be used to project data in order to maximize its classification across classes [34]. This is a cheap and effective alternative, equivalent to performing classification itself. Kernelised versions of discriminant analysis provide simple and convenient methods for non-linear feature extraction and have also been applied to gene expression data [56].

So what are the difficult issues in feature selection and what advice can be offered in this difficult area? First of all, experimental studies should fully understand the nature of their data before applying a method of feature selection or extraction. Adequate tests on feature redundancy in data [57], data normality

[58], noise in data, and overall classification complexity of the task [47,48] should be performed before attempting to select features and perform classification. Some of the challenges with feature selection methods include the determination of probability of error computation when the data distribution means are similar but covariance matrices differ and the estimation of covariance matrix with limited training data [59]. Feature selection techniques have been often accused of as being insensitive to the interaction between different features when selecting the best set. An ideal solution of exhaustive search is, for most practical problems, impossible. Also, if the best feature vector or the best set of feature vectors is not in the direction of the original feature vectors, more features may be needed to achieve the same performance [60]. It is also possible that with limited data, estimates of classification complexity or construction of a meaningful validation set are impossible. In these cases, feature extraction provides a cheaper and better solution in terms of generating a map between a higher dimension space to a lower dimension space such that discriminatory features can have larger weights than non-discriminatory features.

3.3. Missing data

Fig. 1(h) shows ten of the studies surveyed worked with data sets containing missing values. All nine analysed gene expression (microarray) data sets. Often this is due to the raw spot intensities not passing some pre-processing filtering step or transformation, e.g. logarithmic transformation of background subtracted raw intensities, possibly due to problematic registration of the image around the spot, a poorly formed spot of probe material or a weak signal. Of the 10 studies that encountered missing values, six imputed values and the remaining four studies ignored them completely. Failure to deal with missing values correctly can lead to loss of information, or introduce biases in an unforeseen manner.

3.3.1. Critical issues and recommendations

Missing data values must be processed to give us a better quality data set. The simplest (but not necessarily the best) approach is to use only those features that have no missing values. An alternative approach is to develop a classification or unsupervised learning system that only makes computations on the known values, and by standardizing (normalizing) the result ignores the effect of the missing variable. This works well when a sample has a small number of missing variable—for a sample having several missing values, it should be excluded completely. Another possibility for handling missing values is to try to *impute* (estimate) them in some way using the values that were observed. The missing values can be, for example, estimated by the summary statistic of the variable in question (e.g. mean for continuous data and mode for categorical data). This may be a reasonable approach for classification tasks but is not recommended for unsupervised learning. Such a method can be applied when it is suspected that individuals belong to separate groups, which is not known a priori to cluster analysis. Iterative procedures can also be used to impute missing values for unsupervised learning. The basic idea is to cluster data that has missing values and assign each data point to a cluster using membership values. Missing values can be imputed from the statistics of data in their cluster. The data is again clustered now using the imputed values and membership values are estimated again, followed by new imputed values. This continues until the membership values no longer change and the final imputed values have been obtained. The method is most successful when the number of missing values is small. However, in some cases there may be larger proportions of missing values, which is a much harder problem to resolve. For gene expression data a number of imputation approaches have been tried, including *k*NN-based methods [61], PCA (and singular value decomposition (SVD))-based methods [61,62], or simple sample averaging [61,6]. The

*k*NN method provides a simple algorithm but robust and effective imputed values [61], whilst the Bayesian PCA approach of Oba et al. [62] yields superior results.

For gene expression data, even if complete measurements values are obtained for every hybridization difficulties can be compounded by errors contained within the data sets. These errors can take a number of forms: (i) bias and noise in the experimental process, (ii) mis-annotated data points or features, and (iii) incomplete understanding of the biological and experimental processes that lead to the generation of the data. Microarray data typically contains both large systematic and non-systematic errors; systematic errors may be identifiable (and therefore removed) under certain assumptions about the biological processes being measured and the effect of non-systematic errors can be reduced or controlled for by providing replicated measurements. Poorly annotated experiments where the exact nature of the experiment and source of RNA is uncertain make comparison and interpretation of results from the machine learning process difficult. Efforts to standardize the experimental meta-data captured will improve this to some degree [63], but cannot completely eliminate poor or missing annotation particularly with regard to construction of the arrays—spotted arrays constructed from clone libraries may have un-sequenced probes or the sequence of probes may be incorrect [64]. We have also identified lack of understanding as a source of error. Whilst very few scientific endeavors provide a perfect and complete understanding of the processes they attempt to explain, our partial knowledge of the biological and experimental processes informs the construction of the various error models used to identify and remove systematic errors in microarray data. If those assumptions are invalid then so may be the normalized data used as input into the machine learning algorithms. It is inevitable that as our understanding improves pre-processing algorithms and processed data sets will change.

3.4. Data clustering

Data clustering is a generic tool in pattern recognition and data analysis. The techniques can be applied to a wide range of bioinformatics problems. Fig. 1(i) shows a large proportion of studies (10/29 in 2003) used unsupervised learning techniques and typically this was clustering of gene expression data. Clustering of gene expression profiles was one the earliest machine learning techniques applied to large scale gene expression data and is still amongst the most commonly used tools even today. Early approaches used hierarchical clustering, with agglomerative clustering, resulting in what are commonly termed 'Eisen heat-maps' [65], being more popular than divisive clustering [45]. Standard non-hierarchical clustering techniques have also been routinely applied to microarray data, e.g. *k*-means clustering [66], self-organizing maps [25]. Hierarchical clustering is relatively straight-forward to use, with little for the user to determine other than choice of metric and linkage criterion. Output from hierarchical clustering is easy and informative to visualize, leading to identification of functionally similar genes or characterization of unclassified samples. Against this, distinct clusters or classes are not necessarily identified and a gene is not assigned as belonging to a particular cluster. In contrast, *k*-means, by construction produces identifiable clusters to which genes are assigned, but with the complication that the parameter k (the number of clusters) must be chosen a priori or decided through some appropriate model selection procedure. Clustering of bioinformatic data can present the researcher with challenges. Recommendations, to help address these challenges, drawn from the pattern analysis and machine learning research literature are given below. For clustering of gene expression data we also recommend the discussions given by Quackenbush [3] and Causton et al. [67].

3.4.1. Critical issues and recommendations

Data clustering is one of the most difficult problems. Without labeled data, the only way to judge the effectiveness of the solutions is to visualize the results and judge whether the clustering process has found the right clusters. What can be humanly visualized as a cluster, requires a subtle sense of spatial reasoning which is mostly absent in statistical models of data analysis. The optimization of data clustering parameters is still quite a difficult problem and we make some recommendations below. In addition, it is often difficult to know how many clusters to choose. For this purpose, it is important to use cluster validity criteria. Once a clustering procedure has been decided upon resampling procedures such as bagging may be used to assess or improve the accuracy of any clustering obtained [68].

Data clustering based on any form of statistical analysis relies heavily on the use of proximity or similarity measures that compute distances between data points. Data can be broadly classified as of type binary, categorical with more than two levels, continuous, and mixed (categorical and continuous). The recommended similarity measures for these types of data are as follows: binary (matching coefficient, Jaccard coefficient [69], Rogers and Tanimoto measure [70], Sokal and Sneath measure [71], and Gower and Legendre measures [72]); categorical data with more than two levels (genetic dissimilarity measure by Jukes and Cantor [73]); continuous data (a number of distance measures including Euclidean, city block, Minkowski, Canberra, Pearson correlation and angular separation); and mixed data (Gower similarity metric [74]). It is also important to note that inter- and intra-group proximity estimates use different methods (e.g. continuous data can use the summed average of distance between data points and their respective group centers, whereas categorical data uses the relative proportions of samples allocated to different classes in computing a measure proposed by Balakishnan and Sanghvi [75].

The choice of appropriate proximity measure is important. In this context, the nature of data should strongly influence the choice of the proximity measure. Under certain circumstances it is useful to regard continuous data as binary data, where there is a clear step change in the variable values. Secondly, the choice of the measure should depend on the scale of data (binary, multi-level categorical, categorical or mixed) as we mentioned earlier. Finally, in some cases the choice is dependent on the clustering method used. Some clustering methods are based only on the rankings of the proximity estimates and not their absolute values, and some proximity measures may be preferable in this context.

The variables used in data clustering can also be weighted depending on their relative importance. Standardization of variables (putting them on the same range based on their variance and range) can be considered as a special case of weighting. These can be either supplied directly or indirectly. Direct methods are based on some a priori *knowledge* of how important the variables are whereas the indirect methods sometimes use weights that are reciprocal of some measure of variance in that feature. Milligan and Cooper [76] studied eight approaches to variability weights and found that weights based on the sample range of each variable are the most effective. However, weighting variables in this indirect manner has several disadvantages since it dilutes the differences between groups. One good solution is to use a measure of within-group variability to standardize variables rather than total variability [77].

Finally, it is important to have a proper cluster validity criteria which are based on mostly minimizing the intra cluster distance and maximizing inter cluster distance. Everitt et al. [77] discuss several of these criteria. Unfortunately, different criteria produce different results and most of them are statistically motivated without any regard to data structure and user expectation. For this reason, several techniques have been shown to fall apart on clusters that have a meaningful structure even though data points from the same cluster are further apart than between clusters. Model-based techniques measuring cluster validity are lacking, and need further research.

3.5. Experimental design—cross-validation

Although the majority of studies employed some sort of validation scheme, the greatest proportion of studies used a bespoke or non-standard approach to validation, e.g. leave-one-out, 10-fold cross-validation (Fig. 1(j)). A validation scheme constructed specifically for analysis of a particular data set may be more intuitive, but may not validate the machine learning algorithm in well-understood fashion. The performance of standard validation schemes is known and can give the researcher greater confidence in the final validated system.

3.5.1. Critical issues and recommendations

It is important to note that the traditionally used cross-validation method, with the basic error counting scheme (count the number of samples correctly classified) is not the only available technique for the performance estimation of classifiers. Raudys and Jain [31] describe the following approaches to the design of training and test sets, and evaluating classifier performance.

3.5.2. Designing training and test sets

(1) The *re-substitution method* $\Re$: All observations are used to design the classifier and used again to estimate its performance.
(2) The *hold-out method* H: If we have the total number of available observations as n, then a small number of observations N can be used to train the classifier and the remaining $(n - N)$ observations can be used for testing the classifier.
(3) The *cross-validation method C*: In this method, $\binom{n}{k}$ classifiers are designed. Each classifier is designed by choosing k out of n observations and its error rate is estimated using $n - k$ observations. This process is repeated for all distinct choices of k patterns, and the average of the error rates is computed. A popular choice is $k = 1$, which is also known as leave-one-out method.
(4) The bootstrap method β: A bootstrap design sample of size n is formed from the n observations by sampling with replacement. The classification rule is designed using the bootstrap sample and tested twice: (a) n observations of the bootstrap design sample are used to obtain the bootstrap re-substitution estimate P_{R}^{β}, and (b) the original design set is used to obtain the bootstrap estimate of conditional error P_n^{β}. This procedure is repeated r times (typically r lies between 10 and 200). An arithmetic mean of the differences is used to reduce the bias of the re-substitution estimate.

3.5.3. Estimation classification error

There are four different error functions can be used:

(1) *Error counting*, *EC*: This is the typical scheme where the output below a threshold is hard thresholded to belong to one class, otherwise to the other class. For multiple classes, the winner-takes-all method may be used.
(2) *Smooth modification of error counting*, *SM*: The part of the correctly classified observations contribute to the estimation of misclassification probability.
(3) *Posterior probability estimate*, *PP*: This is the probability of a sample belonging to a class. An advantage of this estimate is that the test data can be unlabelled.

(4) *Quasi-parametric estimate*, *QP*: Here it is assumed that the values of the discriminant function have a Gaussian distribution. Error rate is found analytically from sample means and variances of the output of discriminant function for different observations.

Hence, by combining the different options above, we can have 16 different methods of error estimation. Raudys and Jain [31] make the following recommendations for choosing them:

- The re-substitution method results in optimistically biased estimates of asymptotic error rates. Hence, it should only be used when the sample size is large.
- The hold-out error counting estimate results in an unbiased estimate of the expected error rates. The disadvantage of this method is that not all observations of the design sample take part in the learning process and only a part of them are used for calculating classification error.
- The leave-one-out estimate produces a practically unbiased estimate of the expected error rate if the sample observations are statistically independent. For dependent observations, the estimate approaches that of re-substitution method. The main disadvantage is that for some classifiers it is extremely computationally expensive.
- Bootstrap methods and their variants appear to be more accurate than leave-one-out estimates only when the classification error is large.
- The variance of SM, PP and QP estimates can be less than the variance of the EC estimate. The first three estimates are also biased depending on data type.

3.6. Machine learning and experimental design problems

One of the critical issues in applying machine learning for solving bioinformatics problems is the user's ability in understanding the machine learning algorithms, their characteristics and parameter settings. A good experimental design for data analysis and adequate optimization of algorithms is critical to the successful application of machine learning techniques. In this section we briefly discuss some of the commonly occurring difficulties and make recommendations on how to solve them.

One of the major problems is the parameter optimization of different machine learning algorithms used in bioinformatics. Classifiers, clustering methods, missing data imputation techniques, and feature selection methods use a number of parameters that must be optimized. Optimization can be a tedious task especially when these parameters are continuous variables. A general practice for parameter optimization is to use a validation set on which the impact of tuning parameters can be judged and optimized. One of the underlying assumptions of this process is that the validation set closely mirrors the test set, which is often found to be unreliable. On the experimental design, some of the critical problems relate to the amount of data that is necessary for building a reliable and robust machine learning system, how to sample for a validation set, how to choose classifiers (open vs. closed boundary), how to describe a cost matrix and a rejection threshold, how to develop machine learning systems that can automatically determine the optimal parameters.

3.6.1. Recommendations

It is difficult to give a figure on the ratio of the amount of samples and data dimensionality to have a meaningful system. Less number of samples overfit the solutions to these data points. The second important thing is to sample the validation set carefully. In most classification tasks, test data points that

lie within the boundary of the training data are easily classified but outliers and data points that are in the overlap region between two classes are particularly difficult to classify. The validation set must contain such difficult data points as this will develop a system with reasonable robustness against difficult test data. Parameters can be better optimized using a set that works well on a cross-validation task, where for *N*-fold cross validation, the data set is split as *N*-1 folds for training and 1 fold for validation set. This is possible when an extra set of test set is to be used. However, if all that one has is one large data set, then it should be split as training, validation and test for different fold (say 70%, 20% and 10%, respectively). Ignoring all test folds, parameters can be optimized on the average results across training data and their respective validation sets.

Choosing classifiers is a difficult task. Attempts have been made recently to try and understand which classifiers are best suited for what type of data. Sohn [78] performed extensive linear regression analysis with data features as input (dimensionality, sample size, number of classes, data type (categorical vs. continuous), etc.) and the suitability of classifiers as output based on the test accuracy. On the basis of this, a set of simple rules can be derived which can guide us to select the most appropriate classifiers. However, in general we lack specific guidelines for this purpose. We recommend that:

- The choice of the classifier should be guided by the requirements of the application domain. Machine learning techniques based on rules, case-based reasoning and decision trees generate classification rules that are easy to interpret as opposed to the use of neural networks for example.
- In problems that are linearly solvable, which can be easily visualized with PCA plots, it is advisable to use linear discriminant analysis. However, complex classification tasks should be attempted with appropriate guidelines on ensuring classifier generalization. Tools such as nearest-neighbor method must have appropriate distance measures (e.g. Euclidean distance is useful for data that has a spherical distribution, whereas statistical distance which standardizes distance along different feature axes by their corresponding variance is better for elliptical data distributions), and should be optimized for the number of neighbors used on a validation set. On the other hand, classifiers such as neural networks must be built in a systematic manner recommended by Bishop [79] for valid results.
- The combination of classifiers is an important and emerging field. It has long been realized that no single classifier is accurate on all types of data, and by using a combination of diverse classifiers, better results compared to the single best classifier in the ensemble can be obtained. In this context, a number of different classifier combination approaches have been proposed [80] that are based on probability based output combination. Other important alternatives include the concept of stacked generalization [81], and dynamic selection of classifiers [82]. Stacked generalization builds classifiers on top of classifier decisions to predict which combination of classifier decisions should lead to the desired output. Dynamic classifier selection is based on the concept of matching the quality of data sample with the ability of a classifier to best solve such problems (uses different classifiers for different samples). It is recommended that ensembles of classifiers should ensure their diversity, but at the same time encourage them to produce results in a consistent framework that can be combined (e.g. as probability outputs).

The use of rejection thresholds and cost matrices is also important. Rejection thresholds are used for classification where the confidence in a classifier decision can be thresholded (only samples with high confidence are recognized). As a part of initial data screening, this threshold needs to be set. In addition, the misclassifications for different classes may be treated differentially by penalizing some of them more

than others. This can be based on a cost matrix derived mostly from a priori knowledge of the classes involved. Furthermore, receiver operator characteristic (ROC) curves can be used to plot the true and false positive fractions to determine the best classification threshold for binary problems.

Finally, machine learning techniques have the ability to optimize data learning parameters, automatically using labeled results on training data. We would strongly recommend further research into identifying the relationship between data quality and quantity, and tools that are best suited for analyzing them. This is obviously so far a gray and difficult area but systematic approaches, for example by Sohn [78], have shown encouraging results. In fact experience shows that results are far less influenced by our choice of methods of analysis, but much more by our inability to understand data characteristics and use this knowledge to apply those methods optimally.

4. Conclusions

The use of machine learning for solving bioinformatics problems is a relatively new field compared to the use of pattern recognition and machine learning in other domains. In this paper, it has been our aim to make recommendations for bioinformatics from our more detailed understanding of machine learning in the other application domains. The use of a principled approach towards data analysis for bioinformatics may not have been commonplace in the earliest applications. However, our review of the studies published more recently shows that increasingly researchers are conscious of data analysis related issues and the need for proper experimental design and application of machine learning tools. Randomly selected strategies, e.g. for data splitting, parameter optimization, dealing with missing data, how to train classifiers, etc. are giving way to more principled approaches suggested in literature and which guarantee statistical validity and utility of the research. Only by following a principled approach towards data analysis that involves a statistical understanding of the problem, is one able to generate meaningful results that can be interpreted, repeated and applied to practical problems. It is our expectation that our contribution highlights areas in current research where improvements can be made and makes general recommendations. It is not intended to highlight only difficulties (some of which are insolvable due to physical constraints), but to also suggest the way forward to better research in the future.

References

[1] P. Baldi, S. Brunak, Bioinformatics: The Machine Learning Approach, second ed., MIT Press, Cambridge, MA, 2001.
[2] P.O. Brown, D. Botstein, Exploring the new world of the genome with DNA microarrays, Nature Genetics 21 (suppl. 33) (1999).
[3] J. Quackenbush, Computational analysis of microarray data, Nature Rev. Genetics 2 (2001) 418.
[4] J. Quackenbush, Microarray data normalization and transformation, Nature Genetics 32 (2002) 496.
[5] T.R. Golub, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531.
[6] A.A. Alizadeh, et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature 403 (2000) 503.
[7] J. Khan, et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nature Med. 7 (2001) 673.
[8] M.P.S. Brown, et al., Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Nat. Acad. Sci. USA 97 (2000) 262.

[9] T.S. Furey, et al., Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (2000) 906.
[10] Y. Lee, C.K. Lee, Classification of multiple cancer types by multicategory support vector machines using gene expression data, Bioinformatics 19 (2003) 1132.
[11] M.V. Boland, R.F. Murphy, A neural networks classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells, Bioinformatics 17 (2001) 1213.
[12] S. Ahmad, M.M. Gromiha, NETASA: neural network based prediction of solvent accessibility, Bioinformatics 18 (2002) 819.
[13] J.S. Spicker, et al., Neural network predicts the sequence of TP53 gene based on DNA chip, Bioinformatics 18 (2002) 1133.
[14] A. Kohlmann, et al., Pediatric acute lymphoblastic leukemia (ALL) gene expression signatures classify an independent cohort of adult ALL patients, Leukemia 18 (2004) 63.
[15] S.W. Zhang, et al., Classification of protein quaternary structure with support vector machine, Bioinformatics 19 (2003) 2390.
[16] L. Li, C.R. Weinberg, T.A. Darden, L.G. Pedersen, Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics 17 (2001) 1131.
[17] S.A. Armstrong, et al., MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia, Nature Genetics 30 (2002) 41.
[18] S. Kim, Protein β turn prediction using nearest neighbour method, Bioinformatics 20 (2004) 40.
[19] C.M. Perou, et al., Molecular portraits of human breast tumours, Nature 406 (2000) 747.
[20] L.J. van't Veer, et al., Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 484.
[21] S.M. Dhanasekaran, et al., Delineation of prognostic biomarkers in prostate cancer, Nature 412 (2001) 822.
[22] G. Getz, et al., Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data, Bioinformatics 19 (2003) 1079.
[23] W. Liebermeister, Linear modes of gene expression determined by independent component analysis, Bioinformatics 18 (2002) 51.
[24] R.M. Ewing, J.M. Cherry, Visualization of expression clusters using Sammon's non-linear mapping, Bioinformatics 17 (2001) 658.
[25] P. Tamayo, et al., Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc. Nat. Acad. Sci. USA 96 (1999) 2907.
[26] D.C. Hoyle, M. Rattray, PCA learning for sparse high-dimensional data, Europhys. Lett. 62 (2003) 117.
[27] R. Thomson, C. Hodgman, Z.R. Yang, A.K. Doyle, Characterising proteolytic cleavage site activity using bio-basis function neural network, Bioinformatics 19 (2003) 1741.
[28] D.J. Hand, Recent advances in error rate estimation, Pattern Recogn. Lett. 5 (1986) 335.
[29] A.K. Jain, B. Chandrasekharan, Dimensionality and sample size considerations in pattern recognition practice, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Handbook of Statistics, vol. 2, North Holland, Amsterdam, 1982, p. 835.
[30] M. Siotani, Large sample approximations and asymptotic expansions of classification statistics, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Handbook of Statistics, vol. 2, North Holland, Amsterdam, 1982, p. 61.
[31] S. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Machine Intell. 13 (1991) 252.
[32] S. Dudoit, J. Fridlyand, T.P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Ass. 97 (2002) 77.
[33] N. Pochet, F. De Smet, J.A. Suykens, B.L. De Moor, Systematic benchmarking of microarray data classification: assessing the role of nonlinearity and dimensionality reduction, Bioinformatics 20 (2004) 3185.
[34] D. Hwang, et al., Determination of minimum sample size and discriminatory expression patterns in microarray data, Bioinformatics 18 (2002) 1184.
[35] S. Mukherjee, et al., Estimating dataset size requirements for classifying DNA microarray data, J. Comput. Biol. 10 (2003) 119.
[36] J. Cohen, Statistical Power Analysis for Behavioral Sciences, second ed., Erlbaum, Hillsdale, 1988.
[37] H.C. Kraemer, S. Thiemann, How many subjects, Statistical Power Analysis in Research, Sage, CA, 1987.
[38] A.E. Mace, Sample-size Determination, Krieger, Huntington, New York, 1974.
[39] C. Adcock, Sample size determination: a review, Stat. 46 (1997) 262.

[40] I. Guyon, et al., What size test set gives you good error estimates, IEEE Trans. Pattern Anal. Machine Intell. 20 (1998) 52.
[41] C. Cortes, L. Jackel, S. Solla, V. Vapnik, S. Denker, Asymptotic values and rates of convergence, Advances in Neural Information Processing Systems VI, Morgan Kaufmann, Los Altos, CA, 1994.
[42] E.P. Xing, R.M. Karp, CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts, Bioinformatics 17 (Suppl. 1) (2001) 306.
[43] E.P. Xing, M.I. Jordan, R.M. Karp, Feature selection for high-dimensional genomic microarray data, in: Proceedings of the 18th International Conference on Machine Learning, 2001.
[44] M. Xiong, X. Fang, J. Zhao, Biomarker identification by feature wrappers, Genome Res. 11 (2001) 1878.
[45] U. Alon, et al., Broad patterns of gene expression revealed by clustering analysis of tumour and normal colon tissues probed by oligonucleotide arrays, Proc. Nat. Acad. Sci. USA 96 (1999) 6745.
[46] A.K. Jain, D. Zongker, Feature-selection: evaluation, application, and small sample performance, IEEE Trans. Pattern Anal. Machine Intell. 19 (1997) 152.
[47] S. Singh, Multi-resolution estimates of classification complexity, IEEE Trans. Pattern Anal. Machine Intell. 25 (2003) 1534.
[48] S. Singh, PRISM—a novel framework for pattern recognition, Pattern Anal. Appl. 6 (2003) 131.
[49] J. Mao, K. Mohiuddin, A.K. Jain, Parsimonious network design and feature selection through node pruning, in: Proceedings of 12th ICPR, 1994, pp. 622.
[50] B. Yu, B. Yuan, A more efficient branch and bound algorithm for feature selection, Pattern Recogn. 26 (1993) 883.
[51] K. Fukunaga, W.L.G. Koontz, Application of the Karhunen-Loève expansion to feature selection and ordering, IEEE Trans. Comput. 19 (1970) 311.
[52] B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299.
[53] M. Hubert, S. Engelen, Robust PCA and classification in biosciences, Bioinformatics 20 (2004) 1728.
[54] S. De Backer, A. Naud, P. Scheunders, Non-linear dimensionality reduction techniques for unsupervised feature extraction, Pattern Recogn. Lett. 20 (1998) 711.
[55] J.W. Sammon, A nonlinear mapping for data structure analysis, IEEE Trans. Comput. 18 (1969) 401.
[56] J.H. Cho, D. Lee, J.H. Park, I.B. Lee, Gene selection and classification from microarray data using kernel machine, FEBS Lett. 571 (2004) 93.
[57] R.P. Heydorn, Redundancy in feature extraction, IEEE Trans. Comput. 20 (1971) 1051.
[58] G. Brys, M. Hubert, A. Struyf, A robustification of the Jarque Bera test of normality, in: J. Antoch (Ed.), COMPSTAT 2004 Symposium, Springer, New York, 2004.
[59] J.P. Hoffbeck, D.A. Landgrebe, Covariance matrix estimation and classification with limited training data, IEEE Trans. Pattern Anal. Machine Intell. 18 (1996) 763.
[60] C. Lee, D.A. Landgrebe, Feature extraction based on decision boundaries, IEEE Trans. Pattern Anal. Machine Intell. 15 (1993) 388.
[61] O. Troyanskaya, et al., Missing value estimation methods for DNA microarrays, Bioinformatics 17 (2001) 520.
[62] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, S. Ishii, A Bayesian missing value estimation method for gene expression profile data, Bioinformatics 19 (2003) 2088.
[63] A. Brazma, et al., Minimum information about a microarray experiment (MIAME)—towards standards for microarray data, Nature Genetics 29 (2001) 365.
[64] E. Marshall, Affymetrix settles suit, fixes mouse chip, Science 291 (2001) 2535.
[65] M.B. Eisen, et al., Cluster analysis and display of genome-wide expression patterns, Proc. Nat. Acad. Sci. USA 95 (1998) 14863.
[66] S. Tavazoie, et al., Systematic determination of genetic network architecture, Nature Genetics 22 (1999) 281.
[67] H.C. Causton, J. Quackenbush, A. Brazma, Microarray Gene Expression Data Analysis, Blackwell Publishing, Oxford, 2003.
[68] S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, Bioinformatics 19 (2003) 1090.
[69] P. Jaccard, Nouvelles recherches sur la distribution florale, Bulletin de la Societe Vaudoise de Sciences Naturelles 44 (1908) 223.
[70] D.J. Rogers, T.T. Tanimoto, A computer program for classifying plants, Science 132 (1960) 1115.
[71] R.R. Sokal, P.H. Sneath, Principles of Numerical Taxonomy, Freeman, London, 1963.
[72] J.C. Gower, P. Legendre, Metric and Euclidean properties of dissimilarity coefficients, J. Classif. 5 (1986) 5.

[73] T.H. Jukes, C.R. Cantor, Evaluation of protein molecules, in: H.N. Munro (Ed.), Mammalian Protein Metabolism III, Academic Press, New York, 1969, p. 21.
[74] J.C. Gower, A general coefficient of similarity and some of its properties, Biometrics 27 (1971) 857.
[75] V. Balakishnan, L.D. Sanghvi, Distance between populations on the basis of attribute data, Biometrics 24 (1968) 859.
[76] G.W. Milligan, M.C. Cooper, A study of standardisation of variables in cluster analysis, J. Classif. 5 (1988) 181.
[77] B.S. Everitt, S. Landau, M. Leese, Cluster Analysis, Hodder Arnold, London, 2001.
[78] S.Y. Sohn, Meta analysis of classification algorithms for pattern recognition, IEEE Trans. Pattern Anal. Machine Intell. 21 (1999) 1137.
[79] C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
[80] J. Kittler, Mathematical methods of feature selection in pattern recognition, Int. J. Man-Machine Stud. (1975) 609.
[81] S. Dzeroski, B. Zenko, Is combining classifiers with stacking better than selecting the best one, Machine Learn. 54 (2004) 255.
[82] G. Giacinto, F. Roli, A theoretical framework for dynamic classifier selection, in: Proceedings of the 14th ICPR, 2000, pp. 2008.

Harish Bhaskar, graduated from the University of Madras, India with a Bachelor of Technology degree in Information Technology. With a university rank for his undergraduate degree, he joined the University of Exeter in the year 2002 for his Master of Sciences degree in Autonomous Systems. Working in the area machine learning, Harish performed his research work at Motorola UK. Earning a distinction for his Masters degree, in April 2004, Harish joined the ATR laboratory for research in Bio-imaging. He is currently working in the area of live cell imaging. His research interests include, bio-imaging, reinforcement learning, multi-objective learning, medical image processing, dimensionality analysis, autonomous systems and machine learning.

Dr. David Hoyle gained his B.Sc. degree in Mathematics and Physics from the University of Bristol in 1990, and a Ph.D. in Theoretical Physics also from the University of Bristol in 1995. He is currently a Lecturer in Computer Science at the University of Exeter, with research interests in bioinformatics, machine learning and the application of methods from statistical physics to machine learning problems.

Sameer Singh was born in New Delhi, India and graduated from the Birla Institute of Technology, India with a Bachelor of Engineering degree with distinction in Computer Engineering. He received his Master of Science degree in Information Technology for Manufacturing from the University of Warwick, UK and a Ph.D. in Speech and Language Analysis of stroke patients from the University of the West of England, UK. He is currently the Professor of Autonomous Systems and Director of Research School of Informatics at Loughborough University. His main research interests are in image processing, medical imaging, neural networks and pattern recognition. He serves as the Editor-in-Chief of the Pattern Analysis and Applications journal by Springer, Editor-in-Chief of the Springer book series on Advances in Pattern Recognition, and Chairman of the British Computer Society Specialist group on Pattern Analysis and Robotics. He is an Associate Editor of journals including IEEE Transactions on Systems Man and Cybernetics B, Pattern Recognition, Real-time Imaging, Knowledge and Information Systems, and Neural Computing and Applications. He is a Fellow of the Royal Statistical Society, and a Member of BMVA-IAPR, IEE and IEEE.