Neural Comput & Applic (2001)10:39–47

Neural
Computing
& Applications

A Comparison of State-of-the-Art Classification Techniques with Application to Cytogenetics

Boaz Lerner and Neil D. Lawrence

Computer Laboratory, University of Cambridge, Cambridge, UK

Several state-of-the-art techniques – a neural network, Bayesian neural network, support vector machine and naive Bayesian classifier – are experimentally evaluated in discriminating fluorescence in situ *hybridisation (FISH) signals. Highly-accurate classification of valid signals and artifacts of several cytogenetic probes (colours) is required for detecting abnormalities in FISH images. More than 3100 FISH signals are classified by each of the techniques into colour and as real or artifact with accuracies of around 98% and 88%, respectively. The results of the comparison also show a trade-off between simplicity represented by the naive Bayesian classifier, and high classification performance represented by the other techniques.*

Keywords: Bayesian neural network; Fluorescence *in situ* hybridisation (FISH); Multilayer perceptron; Naive Bayesian classifier; Signal classification; Support vector machine

1. Introduction

In recent years, fluorescence *in situ* hybridisation (FISH) has emerged as one of the most significant new developments in the analysis of human chromosomes. FISH offers numerous advantages compared with conventional cytogenetic techniques, since it allows numerical chromosome abnormalities to be detected during normal cell interphase. One of the most important applications of FISH is dot counting, i.e. the enumeration of signals (also called dots) within the nuclei, as the dots in the image represent the inspected chromosomes. Dot counting is used for studying numerical chromosomal aberrations in, for example, haematopoietic neoplasia, various solid tumours, prenatal diagnosis and for demonstrating disease-related chromosomal translocations [1].

However, a major limitation of the FISH technique for dot counting is the need to examine large numbers of cells. This is required for an accurate estimation of the distribution of chromosomes over cell population, especially in applications involving a relatively low frequency of abnormal cells. As visual evaluation by a trained cytogeneticist of large numbers of cells and enumeration of hybridisation signals is expensive and time-consuming, FISH analysis for dot counting can be expedited by using an automatic procedure [2].

One approach to dot counting relies on an auto-focusing microscope to select the 'clearest' image for the analysis [2]. However, basing dot counting on auto-focusing has some shortcomings [3]. Instead, it has been recently proposed [3] to base FISH dot counting on a Neural Network (NN) classifier, discriminating between in and out-of-focus images taken at different focal planes of the same Field-Of-View (FOV), as an alternative to the use of auto-focusing mechanism. Images at different focal planes of a specific FOV compose a stack of images that represents this FOV. Each stack image is analysed, and its signals are classified by the NN as valid data or artifacts, which are the result of out-of-focusing. Following the discrimination of valid signals and artifacts in each stack image, the image that contains no artifacts is selected as the in-focus image to represent the stack (FOV), whereas the other stack out-of-focus images are rejected. The procedure is then repeats itself for other FOVs until the entire slide is covered or the required number of (in-focus) images (or nuclei) are collected. Pro-

Correspondence and offprint requests to: B. Lerner, Computer Laboratory, University of Cambridge, Cambridge CB2 3QG, UK. Email: boaz.lerner@cl.cam.ac.uk

portion estimation of the number of cells having specific numbers of signals can be then performed using these images as in auto-focusing-based dot counting methods [2]. The suggested method enables overcoming most of the shortcomings of auto-focusing, since it does not depend upon a single image. Moreover, the method shortens the length of image acquisition, as stack images are captured coarsely without the necessity to find the exact location of the in focus image. Combining with multi-spectral analysis (Section 2.2), the suggested methodology also shortens the length of image analysis. However, as the system is required to classify valid signals and artifacts, its ability to discriminate between focused and unfocused signals should be more accurate than that of the discriminating element of a system employing an auto-focusing mechanism, as the latter encounters only valid signals. Therefore, the proposed methodology depends upon two components: well-discriminating features to represent valid and artifact signal data and a highly-accurate technique to classify the signals.

Our previous work has investigated feature representation for FISH signals [4] and the application of an NN to signal classification [3]. The suggested methodology enables the replacement of the NN by any other classifier. Therefore, we experimentally compare in the present work state-of-the-art classifiers in discriminating valid and artifact signals of two fluorophores (colours) in order to find the most accurate classification technique. To facilitate the task, we divide the classification procedure into two: classification of signals into colour; and classification of signals as 'real' and 'artifact'. In both cases, two-class classifiers are sought.

Section 2 of the paper describes stages of FISH image analysis that precede signal classification. Section 3 presents the four two-class trainable classifiers: a neural network, Bayesian neural network, support vector machine and naive Bayesian classifier that are evaluated for FISH signal classification. The experimental study is summarised in Section 4, while conclusions for this work are drawn in Section 5.

2. FISH Image Analysis

2.1. Image Acquisition

The process of preparing, hybridising and screening FISH samples, as well as the procedure of capturing FISH images, were described in Lerner et al. [3]. A total of 400 images were collected from five slides and stored in TIFF (Tagged Image File Format) format. An example of a FISH image used for dot counting is shown in Fig. 1, where red and green fluorophores (signals), corresponding to chromosomes 21 and 13, respectively, are seen on blue stained nuclei.

2.2. FISH Colour Image Processing and Segmentation

By analysing each of the three colour channels – Red, Green and Blue (RGB) – of a FISH image separately, image processing and segmentation can be facilitated [3] compared with the conventional intensity-based FISH image analysis [2]. Nuclei can be analysed using the blue channel of the RGB image, whereas red and green signals are analysed in the red and green channels, respectively. Preliminary segmentation on each of the three channels using global thresholds yields the image nuclei and red and green signals. Noise elimination and boundary smoothing of nuclei, as well as spatial correlation between nuclei and signals, complete the segmentation [4]. Finally, colour image analysis does not only facilitate pre-processing and segmentation, but it also yields hue-based features, which are found very efficient for FISH signal representation and classification [4]. Furthermore, it allows the analysis of multiple probes.

2.3. Signal Feature Measurement

Following segmentation, signals are characterised by sets of pixel intensities. A set (signal) can include

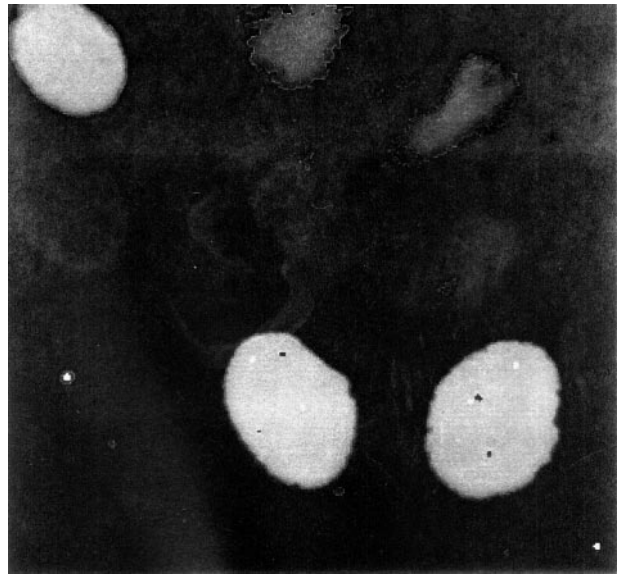

Fig. 1. An example of a FISH image used for dot counting.

one or many members (contiguous pixels). Since the content and dimension of each set can vary dramatically from signal to signal, raw data (intensities) are not considered discriminating enough to act as features for classification. It is therefore necessary to determine a more discriminating and compact representation of the data. One representation can be derived by measuring a set of features of the signal. The features include area (a size measure) and eccentricity (a shape measure), which have been previously suggested [2]. In addition, we measure a number of spectral features [4]. We compute, at the specific colour plane, three RGB intensity-based measurements: the total and average channel intensities and the channel intensity standard deviation. Following the conversion of RGB to HSI (hue, saturation, intensity) colour format, we can also compute four HSI-based measurements: maximum hue, average hue, hue standard deviation, and delta hue. Delta hue is the difference between the maximum and average hue normalised by the average hue. This feature has been added to the set because it was observed that the difference between values of the average and maximum hue for real signals is usually near zero, whereas for some kinds of artifacts (e.g. overlap of two different fluorophores) this difference is substantially large. Two additional features of the set are the two coordinates of the eigenvector corresponding to the largest eigenvalue of the red and green intensity components of the signal. The last feature is the signal average grey intensity ($I_1 = (R + G + B)/3$, where R, G and B are the intensities in the red, green and blue channels, respectively). More details about these features and a motivation for their selection are given in Lerner et al. [4]. Table 1 lists the twelve features used in the work.

Table 1. The set of features studied in the work. Texture indicates standard deviation of the channel intensity (5) or hue (8). Eigenvector 1, 2 are the two coordinates of the eigenvector corresponding to the largest eigenvalue of the red and green intensity components of the signal.

No.	Feature	No.	Feature
1	Area	7	Average hue
2	Eccentricity	8	Hue texture
3	Total channel intensity	9	Delta hue
4	Average channel intensity	10	Eigenvector 1
5	Texture	11	Eigenvector 2
6	Maximum hue	12	Average grey intensity

3. An Overview of Several State-of-the-Art Two-Class Classifiers

Consider a training dataset D which consists of N data points with binary class labels $\{t_1 \dots t_N\}$ and vectors of inputs $\{\mathbf{x}_1 \dots \mathbf{x}_N\}$. We assume that the data was generated by some true underlying function $y(\mathbf{x})$. Our objective is to learn the parameters $\boldsymbol{\theta}$ of some approximating function $f(\mathbf{x}, \boldsymbol{\theta})$ whose form is dependent on our model choice, $\mathcal{M}$, so that we may make 'good predictions' about our class labels.

3.1. Neural Networks

For a Neural Network (NN), we choose a model which predicts the posterior probability of class membership. We define a *likelihood* function as [5]

$$p\,(D|\boldsymbol{\theta}) = \prod_{n=1}^{N} f(\mathbf{x}_n,\boldsymbol{\theta})^{t_n}\,[1 - f(\mathbf{x}_n,\boldsymbol{\theta})]^{(1-t_n)} \tag{1}$$

The approximating function f is represented by the output of an NN with H hidden nodes in its single hidden layer

$$f(\mathbf{x}_n,\boldsymbol{\theta}) = \sigma\left(\sum_{h=1}^{H} v_h g(\mathbf{u}_h^T\mathbf{x}_n)\right) \tag{2}$$

The parameters $\boldsymbol{\theta}$ have been split into the input to hidden weights represented by H vectors $\mathbf{u}_h$, each vector being the weights that 'fan-in' to hidden node h, and $\mathbf{v}$, the vector of the hidden to output weights, consisting of H elements v_h. We have omitted biases for notational simplicity. We take the activation function g to be a hyperbolic tangent, and $\sigma(\cdot)$ is the logistic sigmoid function

$$\sigma(z) = \frac{1}{1 + \exp\,(-z)} \tag{3}$$

which constrains the output of the network to be between 0 and 1, allowing us to interpret f as the probability $P(C_1|\mathbf{x})$ that an input vector $\mathbf{x}$ belongs to class C_1.

We may now define an 'error function' as the negative log likelihood leading to the cross entropy error function

$$-\ln p(D|\boldsymbol{\theta}) = -\sum_{n=1}^{N} \{t_n \ln(f(\mathbf{x}_n,\boldsymbol{\theta})) + (1 - t_n)\ln(1 - f(\mathbf{x}_n,\boldsymbol{\theta}))\} \tag{4}$$

This error function may be minimised by a gradient-based optimisation method.

By using enough hidden units, we may obtain a training error of zero. However, the resulting net-

work will not generalise well on previously unseen data. We need to resort to a sub-partition of the training data known as a validation set to determine an appropriate number of hidden units. The unseen data are then tested on a network with that number of hidden units.

3.2. Bayesian Neural Networks

In the previous section, we reviewed maximum likelihood learning for NNs. Now we briefly introduce the Bayesian approach for inferring parameters of an NN. In Bayesian learning we take our parameters to be random variables, and we aim to determine their *posterior* distribution* for use in making predictions

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{5}$$

where the likelihood $p(D|\boldsymbol{\theta})$ has already been defined in Eq. (1), and we must specify a *prior* distribution $p(\boldsymbol{\theta})$ for the weights. The use of the prior is the main area of controversy in the Bayesian approach, since it requires an interpretation of probabilities as being equivalent to 'beliefs' [6]. The classical frequency-based definition of probabilities does not allow for such an interpretation. The prior represents our belief about what the weights should be before we see the data. A zero mean Gaussian prior is often used

$$p(\boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{O}, \frac{1}{\alpha}\mathbf{I}\right) \tag{6}$$

where α is a 'hyper-parameter' which specifies the inverse width of the prior. This prior reflects a belief that negative weights are as likely as positive weights, and that smaller weights are more probable than larger weights, leading to smoother functions and better generalisation. Once the prior has been selected the integration in Eq. (5) may be performed. To make predictions, we look at the expected output of the network under the posterior distribution

$$\langle f(\mathbf{x},\boldsymbol{\theta})\rangle = \int f(\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|D)d\boldsymbol{\theta} \tag{7}$$

which is the posterior probability of C_1 for a new input vector $\mathbf{x}$.

Unfortunately for neural networks, the integrals required are non-analytic to compute and we must resort to approximations. There are various approaches, in particular the Laplace approximation [7,8], variational approaches [9–11] and Monte-Carlo sampling [12]. We follow the latter approach which involves obtaining samples $\{\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_S\}$ from the posterior distribution, and using them to make sample-based approximations to the required expectations,

$$\langle f(\mathbf{x},\boldsymbol{\theta})\rangle \approx \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x},\boldsymbol{\theta}_s) \tag{8}$$

In particular, we can employ hybrid Monte-Carlo techniques, which make use of gradient information in the sampling.

3.3. The Support Vector Machine

The support vector machine is a technique for classification and regression which arises from the field of *statistical learning theory*. In this work, we only give a brief overview of the support vector machine (for a more detailed introduction, see Burges [13] and Haykin [14]). We define a 'loss function' L, which is the penalty for a classification mistake. Consider first the *risk functional*, which is the expected value of the loss function under the joint probability of the data

$$R(\boldsymbol{\theta}) = \int L(t,f(\mathbf{x},\boldsymbol{\theta}))p(t,\mathbf{x})dtd\mathbf{x} \tag{9}$$

Ideally, to obtain the best approximation $f(\mathbf{x},\boldsymbol{\theta})$ we would minimise the risk functional with respect to $\boldsymbol{\theta}$. Unfortunately, the probability distribution of the data $p(t,\mathbf{x})$ is unknown. Instead, we can look to a sample-based approximation to Eq. (9) known as the *empirical risk*,

$$R_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} L(t_n,f(\mathbf{x}_n,\boldsymbol{\theta})) \tag{10}$$

where we have assumed that our observed data points are independently drawn from the same distribution. The empirical risk may be minimised to obtain an approximation to $y(\mathbf{x})$. This is equivalent to the maximum likelihood approach for neural networks discussed in Section 3.1.

Statistical learning theory relies on notions of capacity which reflects the number of patterns that a classifier may store. If the capacity of our classifier is not infinite, the value of the empirical risk will converge to that of the risk functional as the number of data points tends to infinity [15]. Furthermore, it is possible to place bounds on the rate of convergence which hold with a certain confidence, nor-

* This posterior distribution should not be confused with the posterior probability of class membership.

mally taken to be 95%. These bounds are functions of the model capacity which is often quantified in terms of the *VC dimension* [16]

$$R(\boldsymbol{\theta}) \leq R_{\text{emp}}(\boldsymbol{\theta}) + R_{\text{struct}}(\mathcal{M}) \tag{11}$$

where R_{struct} is known as the *structural risk* of model $\mathcal{M}$, and is a function which increases with increasing model capacity. The principle of structural risk minimisation is to minimise not only the empirical risk, but also the structural risk, through capacity reduction, to obtain the best classifier. This is the underlying theory of the support vector machine.

In the case of two-class classification, the support vector machine aims to place a separating hyper-plane between the two classes. Naturally, there are cases where a linear decision boundary does not exist, and for these cases we must look to *kernel functions*. The kernel functions allow us to project the data onto a very high-dimensional feature space in which it may be separable by a hyper-plane. This has the side effect of also giving the model a very high capacity which increases the structural risk. Fortunately, we may control this capacity by increasing the margin, thereby decreasing the structural risk. As increasing the margin also decreases the capacity, we select the solution with the maximum margin (see Fig. 2). We may increase the margin further by allowing errors in the training set. If a training error is made, the classifier pays a penalty which is proportional to the extent of the error. The constant of proportionality is often denoted by C. Different solutions will then be found depending on our choice of C (Fig. 3).

A common kernel choice is a radial basis functions. This kernel is based upon a Gaussian, and requires the specification of a width parameter σ. The width parameter and the error penalty C may be set through the use of a validation set.

3.4. The Naive Bayesian Classifier

For problems where the task is to minimise the probability of misclassification, the Naive Bayesian Classifier (NBC) provides a simple and clear method, while still enabling impressive performance. The NBC is termed naive, since it makes use of a simplifying assumption that its observable variables,

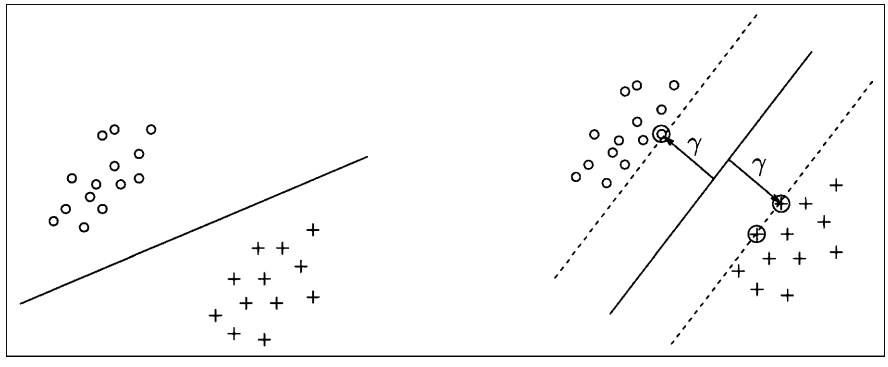

Fig. 2. Two different separating hyper-planes which classify all training examples correctly. The diagram on the right shows the maximum margin. The margin size is denoted by γ. Data points on the margin (circles) are used to define the classifier and are known as support vectors.

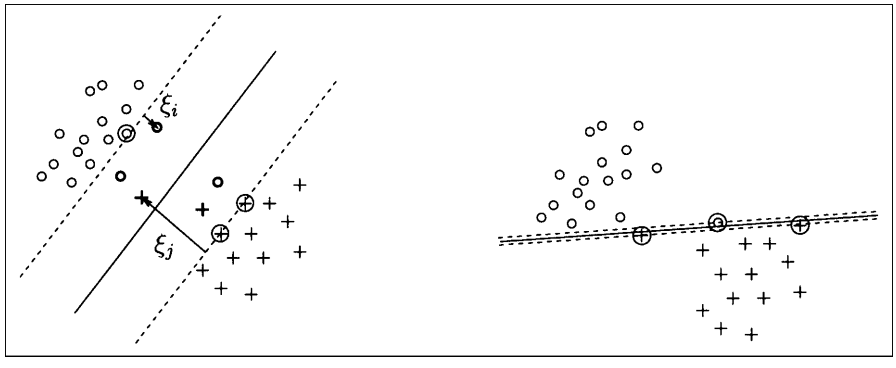

Fig. 3. Depending on the penalty assigned to errors C, different solutions will be found. The left-hand figure is a low error penalty, the right-hand figure is a high error penalty. Bold data points are within the margin. The classifier pays a penalty, $C\xi_i$ ($C\xi_j$), for these data points.

which represent the pattern features, are conditionally independent given the class variable. The classifier can be viewed as a special form of a Bayesian network [17], in which all the edges are directed from the class variable to the observable variables (Fig. 4).

The NBC consists of a finite set $U = \{X_1,X_2,\ldots,X_m,C\} = \{\boldsymbol{X},C\}$ of random variables, where $X_1,\ldots,X_m$ are the observable variables that represent the features, and C is the class variable with K states. The NBC assigns a test pattern $\mathbf{x}$ to the class C_k ($k = 1,\ldots,K$) with the highest *posterior* probability

$$\begin{aligned} P(C_k|\mathbf{x}) &= \frac{p(\mathbf{x}|C_k)P(C_k)}{p(\mathbf{x})} \\ &\propto p(\mathbf{X} = \mathbf{x}|C_k)P(C_k) \\ &= \prod_{i=1}^{m} p(X_i = x_i|C_k)P(C_k) \end{aligned} \tag{12}$$

where $p(\mathbf{x}|C_k)$ is the *class-conditional* probability density, $P(C_k)$ is the *prior* probability for class (C_k), $p(\mathbf{x})$, the *unconditional* density, normalises the posterior probability such that $\Sigma_k\ P(C_k|\mathbf{x}) = 1$, $\mathbf{X} = \mathbf{x}$ represents the event that $X_1 = x_1 \wedge X_2 = x_2 \wedge \ldots \wedge X_m = x_m$ and $\Pi_{i=1}^{m} p(X_i = X_i|C_k)$ is the *likelihood* for $\mathbf{x}$. To derive this equation, we have omitted $p(\mathbf{x})$ which is common to all the states of the class variable and used the NBC independence assumption. Both $P(C_k)$ and $p(\mathbf{x}|C_k)$ can be estimated from the data. $P(C_k)$ is the relative frequency of patterns belonging to C_k out of all the patterns in the data. The data can also be used to estimate $P(x|C_k)$ – the one-dimensional class-conditional probability for discrete variables and $p(x|C_k)$ – the one-dimensional class-conditional probability density for continuous variables. Modelling of probabilities is given by the sample frequency for each value of the variable (that is, the number of times the value is observed divided by the total number of observations). Densities are estimated using different techniques, for example, single density estimation (a parametric method), kernel density estimation (a non-parametric method) or a Gaussian mixture model (a semi-parametric method) [5].

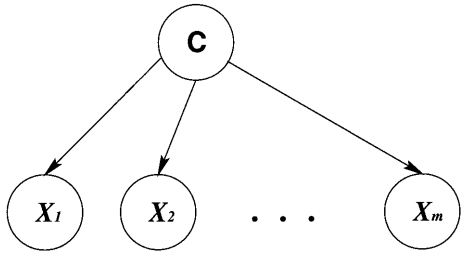

Fig. 4. The naive Bayesian classifier depicted as a Bayesian network in which the observable variables ($X_1,X_2,\ldots,X_m$) are conditionally independent given the class variable (C).

It has been found [18] that Kernel Density Estimation (KDE) of the class-conditional probability densities of the FISH data is more accurate than the other two estimation methods. We model here densities for each class C_k and observable variable X_m using a finite number of data points $\mathbf{x}^n$, $n = 1,\ldots,N_k$, where N_k is the number of training patterns in class C_k. KDE models the one-dimensional class-conditional density as a linear combination of kernel (usually Gaussian) functions

$$p(x|C_k) = \frac{1}{N_k}\sum_{n=1}^{N_k} \frac{1}{(2\pi h^2)^{1/2}} \exp\left\{-\frac{\|\mathrm{x} - \mathrm{x}^n\|^2}{2h^2}\right\} \tag{13}$$

with width h centred around each of the training data points x^n of class C_k.

KDE, as other non-parametric methods, models non-normal distributed data more accurately than parametric techniques, but at the cost of storage and computational complexities, as the number of variables in the model grows linearly with the number of training data points.

4. The Experimental Study

Before beginning the experiments, we established a database of 400 FISH images, which were captured from five slides (Section 2.1). Following nucleus and signal segmentation (Section 2.2), 3144 objects were identified as potential signals and features were measured for them. Based on labels provided by expert inspection (see below), 1145 of the signals were considered as 'reals' (among them 551 were red) and 1999 as 'artifacts' (among them 1224 were red).

Experiments to compare the accuracy of the four classification techniques on signals represented by the twelve features of Section 2.3 (Table 1) were conducted. The normalised ($N(0,1)$) signal features were classified in the first experiment into their colour (red or green), and in the second experiment as real or artifact. Therefore, the input and output spaces in both the experiments were twelve and one-dimensional, respectively. Labels for the patterns, as belonging to each of the classes, were needed to train and evaluate the classifiers, and they were obtained by an expert cytogeneticist using a custom-built graphical environment for labelling FISH images [19].

The first technique, the NN classifier, was a two-layer perceptron trained by the scaled conjugate gradient algorithm [5]. Classification was based on the approximation of the network outputs to the posterior probabilities for the classes. To achieve the highest generalisation capability, the number of hidden units (H) in each classification experiment was determined by the NN configuration achieving the highest accuracy on a validation set. This set was also used to insure that no over-training is performed during the 200 epochs of the training session. Finally, the classifier accuracy was averaged over three random initialisations (committee).

For the Bayesian Neural Network (BNN), the weights were split into four groups: the input to hidden weights, the hidden to output weights, the hidden layer biases and the output layer bias. A hyper-parameter α was associated with each group. For the output layer bias the hyper-parameter was fixed at 1×10^{-4}. The hyper-parameters for the other three groups were treated in a Bayesian manner using a gamma prior specified across α:

$$p(\alpha) = \frac{\alpha^b \alpha^{a-1} \exp(-b\alpha)}{\Gamma(a)} \tag{14}$$

For the input to hidden layer weights, the hidden layer biases and the hidden to output layer weights the parameter a was taken to be 0.25, and the parameter b was taken to be 6.25×10^{-4}, 6.25×10^{-4} and 3.9×10^{-9}, respectively. These values provided very broad hyper-priors, and were used in both the classification experiments. Two hundred samples were drawn from the posterior distribution using hybrid Monte-Carlo sampling [20]. The last 150 samples were used for making predications on the test set. Since a prediction for a test pattern is the integral (average) network prediction over all the models considered under the prior (Eq. (8)), there is no need for validation; any necessary model selection takes place through the Bayesian framework.

For the support vector machine, two parameters needed to be set using a validation set. We considered models with an error penalty C of 1, 10, 100 and 1000, and widths of radial basis function kernels σ of 5, 10, 50, 100, 500 and 1000. These values provided exhaustive ranges of parameters to test the FISH data. We used in the experiments the SVM^{light} implementation of Joachims [21].

Defining a two-class classification task and setting the feature set to include all twelve features, we also determined the NBC structure. By selecting the class and observable variables of the NBC to represent these classes and features, respectively, we employed the NBC for FISH signal classification. Then, we only needed to estimate the class-conditional probability densities for each variable given each of the two states of the class variable. For the variable that represented the signal area, which was the only discrete feature in the set, we modelled the class-conditional probability using the feature sample frequency. For all the other (continuous) features, class-conditional probability densities were modelled using KDE. Different width parameters (h) of the Gaussian kernels that were equal to $T/\sqrt{N_k}$ (where N_k was the number of training data points in C_k and T was in the range 0.01 to 10,000) were checked on the validation set. This choice guarantees that the width parameter shrinks to zero as the number of instances goes to infinity, and density estimation becomes increasingly local as the number of training points increases [17].

The experiments to evaluate the classification accuracy were conducted using the hold out method [22]. This method is applicable for large data sets like the FISH data, which contains more than 3100 data points. We partitioned the data randomly into training and test sets in proportion 70%/30%, respectively. We then selected a model – H, h or C and σ for each of the classifiers – NN, NBC or SVM, respectively, using a validation set drawn from the training set and a five-fold cross-validation (CV-5) experiment. That is, we divided the training set into five disjoint sets using four sets for training the model and the remaining set for validation. We repeated this procedure five times using all the different possible validation sets and averaged the classification accuracy of the five experiments for the model. The procedure was iterated for all the examined models. The model that obtained the highest average classification accuracy on the validation set was selected. This procedure was repeated using the same partitions of the data for all the classifiers (except for the BNN that needed no validation). Figures 5 and 6 present results of model selection experiments performed with the NN and NBC, respectively. The classification accuracy on the training and validation sets is plotted for different models of the two techniques when the patterns are classified into their colour and as 'reals' and 'artifacts'. Following the selection of the 'optimal' model, each of the classifiers was re-trained using its own optimal model and *all* the training data points and then evaluated on the test set.

Table 2 compares the accuracy of each of the four techniques, using its own selected model, in classifying test FISH signals. Signals are discriminated by a classifier of colour (into red or green) and a classifier into real or artifact. The comparison reveals that the BNN is the most accurate technique in both cases (although not always significantly),

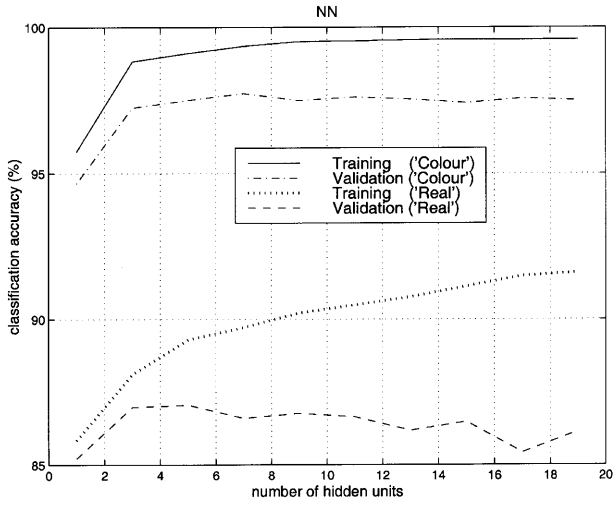

Fig. 5. The classification accuracy of the NN measured on the training and validation sets in the two experiments (classification into colour ('Colour') and into real and artifact ('Real')) for increasing numbers of hidden units. Model selection is based on the highest accuracy on the validation set.

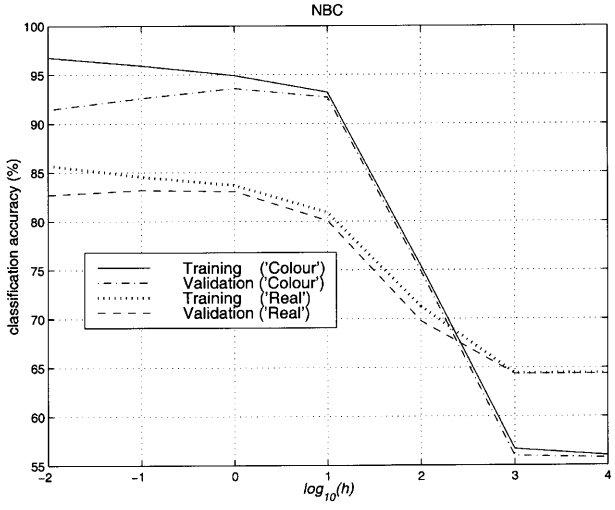

Fig. 6. The classification accuracy of the NBC measured on the training and validation sets in the two experiments (classification into colour ('Colour') and into real and artifact ('Real')) for increasing kernel width parameters (h) (on a log scale). h is equal to $T/\sqrt{N_k}$ (where N_k is the number of training data points in C_k and T is in the range 0.01 to 10,000). Model selection is based on the highest accuracy on the validation set.

and the NN and SVM are comparable and second best. The inferiority of the NBC compared with the other techniques is attributed to the relatively large amount of dependency among features of the set (e.g. average and maximum hue, total and average channel intensities). This dependency violates the independence assumption of the NBC and thereby decreases the classifier accuracy. However, due to their higher complexities, the other techniques can extract additional discriminating information from these correlated features. Finally, the results were compared with those based on a linear classifier [22]. Table 2 shows that a linear classifier is less accurate compared with the top three techniques, but it outperforms the NBC. Although not accurate enough, these two latter techniques provide simplicity which is sometimes vital.

Table 2. Classification accuracies of the four techniques measured on the FISH test set compared with that of a linear model.

Model	Real/artifact (%)	Colour (%)
Neural Network (NN)	86.4	98.1
Bayesian Neural Network (BNN)	88.2	98.8
Support Vector Machine (SVM)	87.2	98.4
Naive Bayesian Classifier (NBC)	83.0	94.0
Linear classifier	84.1	94.6

5. Conclusions

Highly-accurate signal classification is required for precise dot counting in FISH images that are captured without an auto-focusing microscope. To cope with this requirement, we apply in this paper state-of-the-art classification techniques – a neural network, Bayesian neural network, support vector machine and naive Bayesian classifier to the FISH data.

The four trainable classifiers discriminate data based on different approaches. NN training is based on maximum likelihood which is equivalent to the minimisation of an error function. Since the lowest training error does not necessarily represent the model that provides the best generalisation, we are required to draw a validation set from the training set in order to perform model selection. This complicates the experiment and affects the accuracy of the NN as less data is employed for training. Instead of choosing a specific model, the BNN considers a probability distribution function over model space. It uses the data and Bayes' theorem to convert an initial prior distribution for the models to a posterior distribution. This posterior is then used as a weighting function for the predictions made by the network using the different models that are under the prior. The SVM uses capacity control through the maximisation of the margin between the support vectors to improve generalisation performance. However, a validation set is still required to select necessary parameters for this technique. The main difference between the three techniques and the naive Bayesian

classifier is that the latter cannot model correlation between inputs, since it assumes that the observable variables are independent given the class variable. The advantage of kernel density estimation in modelling the class-conditional densities for the NBC is that it is not constrained to any particular functional form.

The first three techniques are found to be highly-accurate compared with the fourth technique and a linear classifier in classifying FISH signals into their colour, as well as into real and artifact. The slight difference in the performance of the BNN and the NN can be attributed to the finite (although large) number of samples leading to different maximum likelihood and maximum *a posterior* solutions. The inferior accuracy of the NBC compared with that of the other techniques can be attributed to the assumption of conditional independence, and to the additional inherent feature extraction stage performed by the other classifiers. However, this inferiority should be weighted against the simplicity offered by the NBC (and also the linear classifier). Moreover, if the features are known to be independent from each other, the simple NBC could be our first choice.

Finally, this research can be extended by evaluating other classifiers for the same data and by comparing the classification techniques on other artificial and real-world databases.

Acknowledgements. The authors would like to thank Seema Dhanjal for data collection and signal labelling and William F. Clocksin for helpful discussions. This work is supported by EPSRC contract GR/L51072: *Automatic Analysis of FISH Images*.

References

1. Tanke HJ, Florijn RJ, Wiegant J, Raap AK, Vrolijk J. CCD microscopy and image analysis of cells and chromosomes stained by fluorescence in situ hybridisation. Histochemical Journal 1995; 27: 4–14
2. Netten H, van Vliet LJ, Vrolijk H, Sloos WCR, Tanke HJ, Young IT. Fluorescent dot counting in interphase cell nuclei. Bioimaging 1996; 4: 93–106
3. Lerner B, Clocksin WF, Dhanjal S, Hultén MA, Bishop CM. Automatic signal classification in fluorescence in-situ hybridisation images. Cytometry 2000 (to appear)
4. Lerner B, Clocksin WF, Dhanjal S, Hultén MA, Bishop CM. Feature representation for the automatic analysis of fluorescence in-situ hybridisation images. Technical Report 464, Computer Laboratory, University of Cambridge, October 1999
5. Bishop CM. Neural Networks for Pattern Recognition, Clarendon Press, 1995
6. Cox RT. Probability, frequency and reasonable expectation. American Journal of Physics 1946; 14: 1–13
7. Gull SF. Bayesian inductive inference and maximum entropy. In: Erickson GJ, Smith CR (eds), Maximum-Entropy and Bayesian Methods in Science and Engineering, Vol 1: Foundations, Kluwer, 1988; 53–74
8. MacKay DJC. A practical Bayesian framework for back-propagation networks. Neural Computation 1992; 4: 448–472
9. Barber D, Bishop CM. Ensemble learning in Bayesian neural networks. In: Bishop CM. (ed), Neural Networks and Machine Learning. Series F: Computer and Systems Sciences 168, Springer-Verlag, 1998; 215–237
10. Hinton GE, van Camp D. Keeping neural networks simple by minimizing the description length of the weights. Proceedings of the Sixth Anuual Conference on Computational Learning Theory 1993: 5–13
11. Lawrence ND, Azzouzi M. A variational Bayesian committee of neural networks. Technical Report, University of Cambridge Computer Laboratory, 1999 (http://www.cl.cam.ac.uk/users/nd121/nnmixture.ps.gz)
12. Neal RM. Bayesian Learning for Neural Networks, Lecture Notes in Statistics 118, Springer-Verlag, 1996
13. Burges CJC. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 1998; 2: 121–167
14. Haykin S. Neural Networks: A Comprehensive Foundation, 2nd ed, Macmillan, 1999
15. Vapnik VN. Estimation of Dependences Based on Empirical Data, Springer-Verlag, 1982
16. Vapnik VN, Chervonenkis AY. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications 1971; 16: 264–280
17. John GH, Langley P. Estimating continuous distributions in bayesian classifiers. In: Besnard P, Hanks S. (eds), Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1995; 338–345
18. Lerner B. A Bayesian methodology and probability density estimation for fluorescence in-situ hybridisation signal classification. Technical Report 474, University of Cambridge Computer Laboratory, October 1999
19. Learner B, Dhanjal S, Hultéu MA. GELFISH – graphical environment for labelling FISH images. Technical Report 465, Computer Laboratory, University of Cambridge, October 1999
20. Neal RM. Software for flexible Bayesian modeling and Markov chain sampling, 1999 (available from http://www.cs.utoronto.ca/ radford/fbm.software.html)
21. Joachims T. Making large-scale svm learning practical. In: Schölkopf B, Burges CJC, Smola AJ. (eds), Advances in Kernel Methods: Support Vector Learning, MIT Press, 1998; 169–184
22. Fukunaga K. Introduction to Statistical Pattern Recognition (2nd ed), Academic Press, 1990