A Primer of Statistical Methods for Classification

Rajarshi Dey and Madhuri S. Mulekar

1 Introduction

Classification is a process of assigning a new subject or item to one of the G known groups or classes on the basis of how closely specific characteristics of this subject/item match with those of the groups. For example, on the basis of specific protein levels measured for a patient, an oncologist can determine with certain confidence whether or not that patient has a certain type of cancer; using a pixel-based satellite image, a geographer can classify land cover into different categories such as water, forested wetland, and upland forest; or using certain admissions criteria, a university can classify applicants as accepted or non-accepted into their program.

There are many different methods (or rules) used to achieve this goal of classification of subjects/items into different classes. Note that classification is not to be confused with clustering as classification involves assigning items to a known number of groups with specific characteristics, whereas clustering involves forming groups of items with similar characteristics when the existing number of groups is unknown. In a world of machine learning and computation, the process of classification is referred to as a supervised learning whereas the method of clustering is an example of unsupervised learning. For example, consider a chicken farm that packages chicken eggs. Before packaging, each egg has to be classified as medium, large, extra-large, or jumbo. This is a case of classification as the classes are well defined based on the egg size and machines can be taught to properly classify eggs based on their size. Sometimes clustering or other pattern recognition methods are

R. Dey · M. S. Mulekar (✉)
Department of Mathematics and Statistics, University of South Alabama, Mobile, AL, USA
e-mail: rajarshidey@southalabama.edu; mmulekar@southalabama.edu

© Springer Nature Switzerland AG 2019

N. Diawara (ed.), *Modern Statistical Methods for Spatial and Multivariate Data*,
STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics & Health,
https://doi.org/10.1007/978-3-030-11431-2_6

used by scientists as a first step towards classification in which existence of number of groups is determined using multiple characteristics of its constituents. In this article, we discuss classification only in the sense of supervised learning, i.e., a procedure in which discriminating variables or functions are used to predict group membership.

Classification into one of the two groups (i.e., $G = 2$) is known as a binary classification and is relatively easier to implement. The earlier classification procedures developed were mostly binary classification methods. But with the development of computing and technology, more literature on different aspects of classification has become available (Hastie et al. 2001).

Broadly speaking there are two types of classifiers: hard and soft. As Wahba (2002) described, the soft classifiers also known as probabilistic classifiers typically provide conditional probability of membership for each of G ($G \geq 2$) groups for each new subject to be classified and then put the subject in the group with the largest probability of membership. In contrast, hard classifiers only provide a hard classification boundary like a fence around a property for each group based on the explanatory variables. A new subject with characteristics within the boundary of a group is assigned to that particular group.

The development of a classification procedure involves two major steps, classification and validation. In the first step, a classification rule or an algorithm also known as a classifier is developed and in the second step, performance of this classifier is evaluated. Dataset used in the first step to develop a classification rule is known as the *training dataset* and includes information about group membership (response or output) of each subject along with other random variables (explanatory variables or features). A similar dataset is used in the second step to validate the classifier developed in the first step and is known as the *validation dataset*.

Let us assume that all outcomes are independently observed for random response variable Y and a vector of random explanatory variables $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ \ldots \ , \mathbf{X}_p)$. Let $\{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots n\}$ indicate a training dataset consisting of n independent measurements. Each measurement is a $(p + 1)$-tuple where $\mathbf{x}_i = (x_{1i}, \ x_{2i}, \ \ldots \ , x_{pi})'$ is a vector of outcomes for p explanatory variables and y_i ($i = 1, 2, \ \ldots, n$) is the outcome for response variable, which only shows membership of ith unit to a certain class.

Consider the famous iris dataset by R.A. Fisher (Anderson 1935) containing 150 data points with four explanatory variables and three classes of iris. The explanatory variables are sepal-length, sepal width, petal-length, and petal-width of three types of irises, namely *setosa*, *virginica,* and *versicolor*. The goal is to identify type of iris using sepal and petal measurements. This is a classification problem. Sepal-length and petal-length distributions in Fig. 1 show that although *setosa* tend to have lowest and *virginica* tend to have highest sepal-lengths the separation among three classes based on sepal-length is not clear due to considerable overlap and hence sepal-length by itself is not a good classifier for these three types of irises. On the other hand, petal-length is clearly able to distinguish *setosa* from the other two but not between *virginica* and *versicolor* possibly resulting in misclassification. Hence there is need for more than one predictor to reduce misclassification. Figure 2

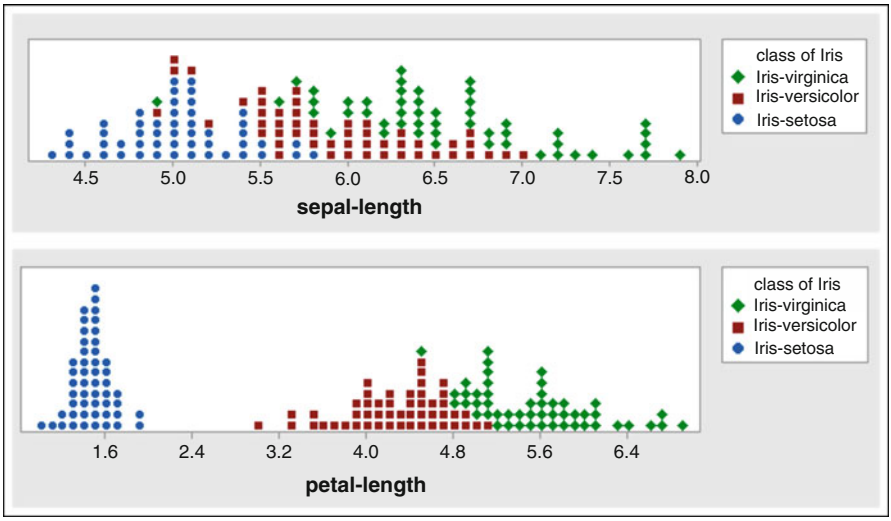

Fig. 1 Distributions of sepal- and petal-lengths of three varieties of iris

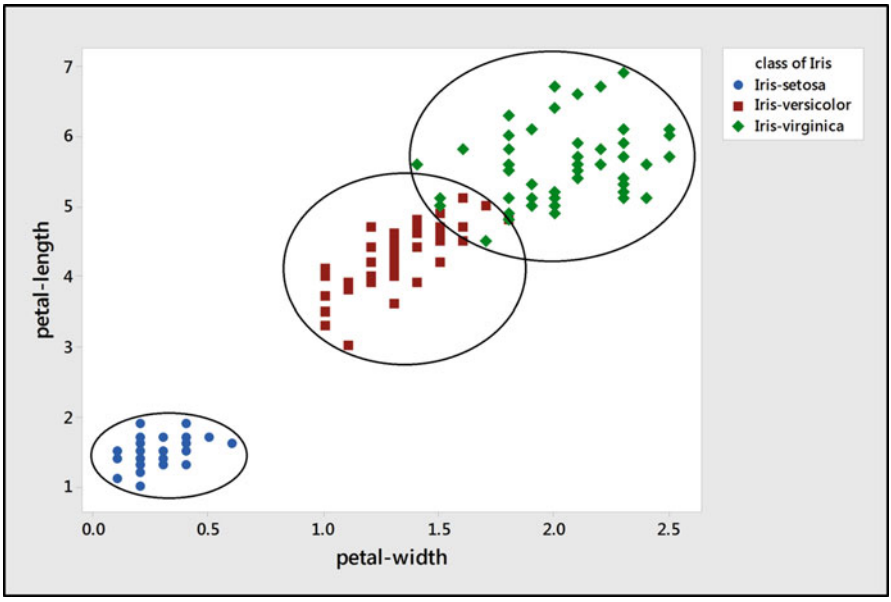

Fig. 2 A scatterplot of iris petal-length vs petal-width

shows a scatterplot of iris petal-length versus petal-width. Once again *setosa* are clearly separated from the other two varieties, but there is still some overlap between *virginica* and *versicolor* possibly leading to misclassification.

In this article, we aim to provide basic description of the most well-known and commonly used classification methods that are used to develop classifiers (or classification rules) based on relation between the response variable Y and explanatory variables $\mathbf{X}$, which then are used to assign new objects to these known groups based on observed $\mathbf{x}_0'$. Two soft classifiers (logistic regression and naïve Bayes estimator) and four hard classifiers (linear discriminant analysis, support vector machines, K nearest neighbor, and classification trees), respectively, are described in Sects. 2 and 3 along with their strengths and weaknesses. Some discussion assessing performance of these classifiers for five different datasets, three real and two simulated, is provided in Sect. 4. Some concluding remarks about choice of classifiers in practice are provided in Sect. 5.

2 Soft Classifiers

Intuitively, a soft classifier should appeal to anyone who likes to incorporate the uncertainty of outcome provided by classifiers because it also shows the likelihood of a new observation being a member of different classes. Here two most commonly used soft classifiers, namely logistic regression and naïve Bayes classifiers are discussed.

2.1 Logistic Regression

As described by Cramer (2003), the first use of logistic function in logistic regression was traced to modeling population growth rate in the nineteenth century Africa. Berkson (1944) suggested the use of logistic probability density function (pdf) instead of normal pdf in certain bioassay procedures. He also coined the term *logit* model to describe the resulting model. Later many researchers in statistics and epidemiology started working on what would eventually become one of the most widely used methods in classification, namely the logistic regression, particularly with $G = 2$ groups. Cox (1969) is considered one of the pioneers in binary logistic regression. More generalized versions of logistic regression, which can classify new items into G ($G \geq 2$) groups, are credited to Gurland et al. (1960), Mantel (1966), and Theil (1969).

For the sake of simplicity, let's start with the case of binary logistic regression with two classes being coded as 1 and 2 (i.e., y_i= 1 or 2). For $\pi_{i1} = P(y_i = 1 | \mathbf{x}_i)$, $i = 1, 2, \ldots, n$ and under the assumption that the response variable y_i has a

Bernoulli distribution with parameters π_{i1}, the logistic model is given by,

$$\pi_{i1} = E\left(y_i = 1|\mathbf{x}_i\right) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ji}\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ji}\right)} \tag{1}$$

where $\beta_0, \beta_1, \ldots, \beta_p$ are $(p+1)$ regression coefficients. Note that in a binary case, $\pi_{i2} = 1 - \pi_{i1}$.

Alternatively this model can be presented as,

$$logit\left(\pi_{i1}\right) = \log\left(\frac{\pi_{i1}}{1-\pi_{i1}}\right) = \beta_0 + \sum_{j=1}^{p}\beta_j x_{ji} \tag{2}$$

The regression parameters $\beta_j, j = 0, 1, \ldots, p$ are estimated from the available training dataset. The maximum likelihood (ML) estimates $\hat{\beta}_j$, $j = 0, 1, \ldots, p$ are obtained by maximizing the likelihood function

$$L\left(\beta_0, \beta_1, \ldots, \beta_p\right) = \prod_{i=1}^{n}\pi_i^{y_i}(1-\pi_i)^{n-y_i}.$$

Since there are no closed-form solutions available for maximizing this likelihood function, iterative algorithms are used to obtain the ML estimates of regression parameters. According to Agresti (2013), the most popular choices for iterative algorithms are either the Newton–Raphson algorithm (Tjalling 1995) or iteratively reweighted least square (IRWLS) algorithm (Burrus et al. 1994). However, sometimes due to the use of too many explanatory variables or highly correlated explanatory variables, these algorithms fail to converge resulting in failure to estimate parameters. Another counter-intuitive situation sometimes occurs when there is a complete separation between two classes using some linear combination of explanatory variables. More information on estimating parameters of logistic regression is available in Menard (2002).

A simple extension of logistic regression from binary to multiclass classification is known as multinomial logistic regression. The multinomial logistic model is given by,

$$\pi_{ig} = \frac{\exp\left(\beta_{0g} + \sum_{j=1}^{p}\beta_{jg}x_{ji}\right)}{1 + \sum_{g=1}^{G-1}\exp\left(\beta_{0g} + \sum_{j=1}^{p}\beta_{jg}x_{ji}\right)}, \quad g = 1, 2, \ldots, G-1 \text{ and } i = 1, 2, \ldots, n. \tag{3}$$

Extending notation used in the binary case to $G \geq 2$ groups, we can write $\pi_{ig} = P(y_i = g|\,\mathbf{x}_i)$, for $i = 1, 2, \ldots, n$ and $g = 1, 2, \ldots, G$. Although it does not matter which category is chosen as baseline, generally category G is used as a

baseline and π_{iG} can be obtained using the fact that $\pi_{iG} = 1 - \sum_{g=1}^{G-1} \pi_{ig}$. From the point of estimation, there are $(p+1)(G-1)$ model parameters to be estimated. For estimating these parameters, the ML estimation or the maximum a posteriori (MAP) methods are commonly used (Murphy 2012). Estimation method MAP is similar to ML in the sense that it chooses that value of parameter which maximizes the value of a mathematical function, in this case the posterior distribution of the parameter itself. Most of the times, a closed-form solution is not available, hence different algorithms are used for estimation and IRWLS is a popular choice among practitioners.

If the $G \geq 2$ classes are ordered using an ordinal response variable, an alternative popular model often used in practice is the proportional-odds cumulative logit model. For example, consider a typical Likert scale question where the responders are asked to grade certain experience on a scale of 1 to 5 with 1 being the worst rating and 5 being the best. It might be of interest to determine if there exist some explanatory variables that can explain how the responders rate their experience. First developed by Snell (1964), this model is given by,

$$L_{ig} = \log \frac{\sum_{c=1}^{g} \pi_{ic}}{\sum_{c=g+1}^{G} \pi_{ic}} = \beta_{0g} + \sum_{j=1}^{p} \beta_j x_{ji}, \quad \text{for } g = 1, 2, \ldots, G \text{ and } i = 1, 2, \ldots, n. \tag{4}$$

Here, L_{ig} represents the log-odds of two cumulative probabilities. A manageable number of total $(G - 1 + p)$ parameters are to be estimated from this model. Typically, ML estimates of parameters of this model are obtained using iterative algorithms such as IRWLS and majorization-minimization (Lange 2016).

2.2 *Naïve Bayes Classifier*

Naïve Bayes (NB) is a family of soft classifiers that uses the Bayes theorem (Bayes 1763) along with a very strong assumption of independence among explanatory variables which is often unrealistic. However, this classifier works very well in the presence of dependencies among many categorical explanatory variables (Rish 2001) and is quite fast to execute even with large datasets.

NB classifier differs from the logistic regression classifier in terms of how the probability π_{ig} is modeled. When using a logistic regression classifier, $\pi_{ig} = P(y_i = g | \mathbf{x}_i)$ is modeled directly from data. On the other hand, when using a Naïve Bayes classifier, first the estimates for $P(\mathbf{x}_i | y_i = g)$ are obtained from data and then assuming independence among explanatory variables, π_{ig} is modeled using Bayes theorem as,

$$\pi_{ig} \propto P(y_i = g) \prod_{j=1}^{p} P\left(x_{ji} | y_i = g\right), \quad g = 1, 2, \ldots, G \text{ and } i = 1, 2, \ldots, n. \tag{5}$$

The estimate for $P(y_i = g)$ can be obtained from the training set as the proportion of training set observations that belong to class g ($g = 1, 2, \ldots, G$). The estimates for $P(x_{ji} | y_i = g)$ are typically obtained via ML estimation technique. A new observation is assigned to a group for which probability π_{ig} is maximum among all G groups.

Estimating parameters from the likelihood function depends on how the likelihood, $P(x_{ji} | y_i = g)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$, and $g = 1, 2, \ldots, G$ is modeled parametrically. If X_j is a continuous random variable, then the popular choice of distribution is normal (Gaussian) such that $(X_j | Y = g) \sim N(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$. If X_j is a categorical random variable with m categories, then the most commonly used distribution is multinomial, i.e., $(X_j | Y = g) \sim Multinomial(1, \phi_{1g}, \ldots, \phi_{mg})$ for one trial where ϕ_{lg}, $l = 1, 2, \ldots m$ is the probability associated with the l^{th} category such that $\sum_{l=1}^{m} \phi_l = 1$.

The NB classifier is remarkably effective considering the assumptions needed to obtain the probabilities are almost always wrong (Hand and Yu 2001). This method is a building block to what is commonly known as a Bayesian spam filter (Nigam et al. 2000) used by the email providers. A semi-parametric version of NB classifier performs much better when the explanatory variables are obviously non-normal (Soria et al. 2011).

3 Hard Classifiers

Hard classifiers typically do not provide a probability of group association. In other words, there is no uncertainty associated with classification because classifier provides a hard boundary between groups and exactly for this reason some researchers like to use them**.** Four commonly used classifiers discussed here are linear discriminant analysis, K nearest neighbor, support vector machines, and classification trees.

3.1 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a hard classification method. Statistical literature indicates that LDA is one of the first methods developed for classification and its basic idea originated from none other than Fisher (1936). The basic idea behind LDA is to determine that linear combination of explanatory variables which will magnify the difference between two classes making it easier to achieve correct classification. The generalization of this idea for classification into $G (G > 2)$ classes is credited to Rao (1948). The NB classifier is similar to LDA in nature (Hand and Yu 2001), although in LDA the aim is obtain a classifier while in NB there is more emphasis on identifying a class with the maximum posterior probability.

Fisher (1936) proposed a classification rule for two groups which involved determining a vector $\mathbf{r}$ that maximizes function $\delta(\mathbf{r}) = (\mathbf{r}' \Sigma \mathbf{r})^{-1} (\mathbf{r}' (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2))^2$

under the assumption that $(\mathbf{X}|\, Y = g) \sim N(\boldsymbol{\mu}_g, \boldsymbol{\Sigma})$ for $g = 1, 2$. This is equivalent to finding a hyperplane that provides a solution to equation

$$\log\left[\frac{P\,(Y = 1|\mathrm{X})}{P\,(Y = 2|\mathrm{X})}\right] = 0. \tag{6}$$

Using Bayes' rule, we can write,

$$P\,(Y = g|\mathbf{X}) = P\,(Y = g)\,\frac{P\,(\mathbf{X}|Y = g)}{P\,(\mathbf{X})} \quad \text{for } g = 1, 2$$

where $p_g = P(Y = g)$ for $g = 1, 2$ is the overall class probability and can be estimated from the training data. Under the assumption that the explanatory variables are multivariate normal, the hyperplane can be found by solving the following equation for $\mathbf{r}$,

$$\log\left[\frac{p_1}{p_2}\right] + \mathbf{r}^{'}\boldsymbol{\Sigma}^{-1}\,(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \frac{1}{2}\left(\boldsymbol{\mu}_1^{'}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^{'}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2\right) = 0. \tag{7}$$

Solution to (7) leads to a *linear* classifier (or a linear boundary between two groups) because Eq. (6) is a linear function of explanatory variables. The first step in LDA is to estimate the mean vectors ($\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$) and variance–covariance matrix ($\boldsymbol{\Sigma}$) from the training dataset. For any new observation, $\mathbf{x}_0$, one can estimate $\Delta_{\mathbf{x}_0}$ from (8) as,

$$\hat{\Delta}_{\mathbf{x}_0} = \log\left[\frac{\hat{p}_1}{\hat{p}_2}\right] + \mathbf{x}_0^{'}\hat{\boldsymbol{\Sigma}}^{-1}\left(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2\right) - \frac{1}{2}\left(\hat{\boldsymbol{\mu}}_1^{'}\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2^{'}\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_2\right). \tag{8}$$

Using this $\hat{\Delta}_{\mathbf{x}_0}$ value a new observation $\mathbf{x}_0$ is assigned to one of the two groups as follows:

$$\text{Assign } \mathbf{x}_0 \text{ to } \begin{cases} \text{Group 1} & \text{if } \hat{\Delta}_{\mathbf{x}_0} > 0 \\ \text{Group 2} & \text{if } \hat{\Delta}_{\mathbf{x}_0} < 0. \end{cases}$$

In cases where the assumption of homoscedasticity of variance–covarinace matrix is not justified and a more general underlying assumption is that $(\mathbf{X}|\, Y = g) \sim N(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ for $g = 1, 2$, a *quadratic* classifier (i.e., a quadratic function) is used to describe a boundary between two classes. This procedure is known as quadratic discriminant analysis (QDA) (Hastie et al. 2001). In QDA, the hyperplane can be obtained by solving (9) for $\mathbf{r}$.

$$\begin{aligned} &\log\left[\tfrac{p_1}{p_2}\right] - \tfrac{1}{2}\log\tfrac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|} + \mathbf{r}^{'}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \\ &\quad - \tfrac{1}{2}\mathbf{r}^{'}\left(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}\right)\mathbf{r} - \tfrac{1}{2}\left(\boldsymbol{\mu}_1^{'}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^{'}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_2\right) = 0 \end{aligned} \tag{9}$$

As can be seen from (7) and (9), a QDA requires more parameters to be estimated from the training dataset, precisely $(2 + 2p + 2p^2)$ parameters for QDA compared to $(2 + 2p)$ for LDA. That can lead to a serious issue if the training dataset is small. To overcome this issue, Srivastava et al. (2007) proposed an effective Bayesian solution.

A simpler method under the assumption of homoscedasticity of variance–covariance matrices is to use Mahalanobis distance (Mahalanobis 1936) for classification. For any new observation, $\mathbf{x}_0$, a linear discriminant function LDF_g is computed for each group (see (10)) under the assumption that $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}$, respectively, are the unknown mean vector and variance–covariance matrix of $\mathbf{X}$.

$$LDF_g(\mathbf{x}_0) = \mathbf{x}_0' \hat{\boldsymbol{\Sigma}}_0^{-1} \hat{\boldsymbol{\mu}}_g - \frac{1}{2} \hat{\boldsymbol{\mu}}_g' \hat{\boldsymbol{\Sigma}}_0^{-1} \hat{\boldsymbol{\mu}}_g + \hat{p}_g \tag{10}$$

where $\hat{\boldsymbol{\Sigma}}_0$ is the pooled estimate of the common variance–covariance matrix $\boldsymbol{\Sigma}$ and $\hat{p}_g$ is the estimate of probability $p_g = P(Y = g)$ for $g = 1, 2, \ldots, G$ obtained from the training data. Then the new observation $\mathbf{x}_0$ is assigned to the group with the highest discriminant function value, i.e., the group corresponding to $\max\{LDF_g(\mathbf{x}_0), g = 1, 2, \ldots, G\}$.

Although LDA is quite effective in many situations (Hand 2006), in some situations the joint pdf of explanatory variables differs considerably from the multivariate normal distribution. In such cases semi-parametric LDA technique derived by Mai and Zou (2015) under the assumption of sparse variance–covariance matrix is more effective.

3.2 K Nearest Neighbor

Assumed to have originated in long past, the history of $K(1 < K < n)$ nearest neighbor (KNN) classification is not really that well known. In modern times, Sebestyen (1962) described this method as *proximity algorithm* and Nilsson (1965) called it the *minimum distance classifier*. Cover and Hart (1967) were the first to name this algorithm as the *nearest neighbor* and that name became popular.

Although mostly used as a hard classifier, KNN can be used as a soft classifier too. The idea behind KNN is quite simple and no parametric assumption is required. Given a training dataset of size $n(n > K)$, this classification algorithm starts when a new observation, $\mathbf{x}_0 = (x_{01}, \ldots, x_{0p})'$, is recorded with known values for all explanatory variables but unknown class. The first step is to calculate the K nearest neighbors in terms of the explanatory variables. Using some well-defined distance measure, distance $d_i = d(\mathbf{x}_0, \mathbf{x}_i)$, $i = 1, 2, \ldots, n$ between this new observation and each observation from the training dataset is calculated and these distances are ordered as $d_{(1)} \leq d_{(2)} \leq \ldots \leq d_{(n)}$. Considering the lowest K distances, $\{d_{(1)}, d_{(2)}, \ldots, d_{(K)}\}$, the class membership of these closest K neighbors in the training dataset is determined. Then the new observation is placed in the class that

has the largest number of these K neighbors. For example, suppose k_g of the nearest K neighbors belong to group $g(g = 1, 2, \ldots, G)$ such that $\sum_{g=1}^{G} k_g = K$, then the new observation is placed in the group c if $k_c = \max\{k_g, g = 1, 2, \ldots, G\}$. Note that there is a possibility that no such unique maximum exists for a given new observation and a chosen K, thus resulting in ties. Although not exactly a group inclusion probability, these nearest neighbors can be used to provide a group membership indicator of the new observation using relative fractions (k_g/K), $g = 1, 2, \ldots, G$.

Now the question is: how to choose value of K, the number of nearest neighbors to be used? Given a large dataset one can always use cross-validation and choose the K value corresponding to the lowest misclassification rate in the validation dataset. Note that choice of a too small value for K indicates that the space generated by the explanatory variables is divided into many small subspaces and the class membership of a new observation depends on which subspace the new observation belongs to. In that case outliers in the original dataset can create problems in predicting the class membership of a new observation that is close to the outlier resulting in a higher variance in prediction. However, choice of a large value for K basically leads to division of the training data space into G smooth subspaces which in turn creates the problem of misclassification of any outlier of these subspaces and subsequently higher bias in prediction. As a rule of thumb, $K = \sqrt{n}$ is considered to be a sensible choice for number of classes in practice. If the number of groups in the data is 2 (i.e., $G = 2$), then K should be an odd number to avoid the possibility of ties in group membership indicators.

The most popular choice for a distance measure is the Euclidean distance which for a new observation, $\mathbf{x}_0 = (x_{01}, \ldots, x_{0p})'$, is calculated as,

$$d_i = \sqrt{\sum_{j=1}^{p} \left(x_{0i} - x_{ji}\right)^2}, \quad i = 1, 2, \ldots, n. \tag{11}$$

Some other distance measures used commonly in practice are Hamming distance (Hamming 1950) and Chebyshev distance (Grabusts 2011). Chomboon et al. (2015) looked at eleven different distance measures and found that the Euclidean, Chebyshev, and Mahalanobis distance measures perform well. For a synopsis of different distance measures, please refer to Mulekar and Brown (2014). Different explanatory variables tend to have different range of possible values and some distance measures such as the Euclidean distance tend to be affected by the range of measurements. Hence in practice, datasets are typically normalized before classification to reduce the influence of explanatory variables with larger range of measurements. When using a dataset with a large number of explanatory variables, to reduce the computation time, a dimension reduction technique such as principal component analysis (PCA) is used. To overcome the problem of choosing a value for K, Samworth (2012) suggested the use of weighted nearest neighbor algorithm in which instead of choosing K nearest neighbors (i.e., essentially assigning a weight

of $1/K$ to K nearest neighbors and 0 to the remaining observations in the training dataset while assigning a class to the new observation), all observations in the training dataset are assigned a weight using some optimal weighting scheme. When dealing with a big dataset, an approximation to the method of nearest neighbors proposed by Har-Peled et al. (2012) is useful.

3.3 Support Vector Machine

Support vector machine (SVM) is a class of hard classifiers. For a binary classification with p explanatory variables, an SVM classifier constructs a $(p-1)$-dimensional hyperplane in the p^{th} dimension to maximize the margin. Here margin refers to the distance between the observation closest to the boundary of a group and the remaining groups. Points on or closest to the boundary of decision surface are called support vectors and they are used in learning models associated with the classification algorithm. The idea behind SVM is to find that hyperplane which provides the maximum margin from support vectors among infinitely many possible hyperplanes that can separate two groups provided the two groups are completely separable. For a binary classification, one hyperplane known as the maximum margin hyperplane is constructed. For $G > 2$ groups, more than one such maximum margin hyperplanes need to be created to separate groups and a combination of these hyperplanes is used for the classification of a new observation.

Consider the case of binary classification, and assume that there actually exists a linear hyperplane of the form

$$W(\mathbf{X}) = w_0 + \sum_{j=1}^{p} w_j X_j \tag{12}$$

that can perfectly differentiate between two classes. Then a method described by Vapnik and Lerner (1963) can be used to find a maximum margin hyperplane. Maximum margin hyperplane is a hyperplane for which $W(\mathbf{X}) = 0$. In SVM, only support vectors obtained using the training data are used to estimate the coefficients of explanatory variables in (12). Since the decision surface differentiates the classes completely, the linear function in (12) should be positive for one group and negative for another. Without any loss of generality, assume that for support vector(s) in group 1, $\hat{W}(X) = -1$ and for those in group 2, $\hat{W}(X) = 1$. In order to maximize the margin, it is sufficient to minimize $\sum_{j=1}^{p} w_j^2$ subject to $v_i \hat{W}(\mathbf{x}_i) \geq 1, i = 1, 2, \ldots, n$ where $v_i = -1$ if $y_i = 1$ and $v_i = 1$ if $y_i = 2$. Thus this hyperplane can be obtained by minimizing the Lagrangian formulation,

$$L = -\sum_{i=1}^{n} a_i \left(v_i W(\mathbf{x}_i) - 1\right)$$

where $a_i(\ i = 1, 2, \ \ldots\ , n)$ are Lagrange multipliers. Once this hyperplane is estimated, $\hat{W}(\mathbf{x}_0)$ is computed for any new observation $\mathbf{x}_0$ and the new observation is assigned to the group 1 if the $\hat{W}(\mathbf{x}_0) < 0$ and to group 2 if $\hat{W}(\mathbf{x}_0) > 0$.

In many practical situations, a perfectly differentiating hyperplane does not exist. For such situations, Cortes and Vapnik (1995) proposed a modification to the maximum margin hyperplane to differentiate between two groups. They proposed estimating the hyperplane with the help of a hinge loss function, $h(\mathbf{x}) = \max\left(0, 1 - v\hat{W}(\mathbf{x})\right)$. Note that unlike a linearly separable case where $v_i \hat{W}(\mathbf{x}_0) \geq 1\ \forall i$; for linearly non-separable cases, the possibility of $v_i \hat{W}(\mathbf{x}_0) < 0$ exists for a few support vectors. So, the hinge loss function is 0 for such support vectors and this is used to penalize such support vectors while estimating the decision boundary. Thus, instead of minimizing $\sum_{j=1}^{p} w_j^2$ subject to $v_i \hat{W}(\mathbf{x}_i) \geq 1, i = 1, 2, \ldots, n$, function

$$\theta \sum_{j=1}^{p} w_j^2 + \frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - v_i \hat{W}(x_i)\right) \tag{13}$$

is minimized where θ is a penalty parameter. This is known as the *soft* version of SVM, although this is not a soft classifier.

Since a linear classifier does not always exist, researchers extended the idea of SVM to find a non-linear classifier. Using ideas first promoted by Aizerman et al. (1964), Boser et al. (1992) proposed the use of kernelization to obtain a non-linear classifier by improving a classifier obtained via SVM. The trick is to use a transformation $\mathbf{Z}$ of $\mathbf{X}$ such that the new transformed explanatory variables $\mathbf{Z}(\mathbf{X})$ provide a better classifier than the one provided by $\mathbf{X}$. Then, proceed to obtain an SVM based on the new transformed explanatory variables, $\mathbf{Z}$. A carefully chosen transformation $\mathbf{Z}$ can possibly result in a linear classifier. Note that $\mathbf{Z}$ is not observed or calculated from data but it is replaced by the kernel function, κ, such that $\kappa(\mathbf{x}_i, \mathbf{x}_l) = \mathbf{z}_i' \mathbf{z}_l$ for $i \neq l = 1, 2, \ \ldots\ , n$. There are many kernel functions in use, but the most used Gaussian kernel (Schölkopf et al. 1997) is given by,

$$\kappa(x_i, x_l) = \exp\left(-\xi \sum_{j=1}^{p} \left(x_{ij} - x_{lj}\right)^2\right).$$

To develop a multiclass SVM classifier (i.e., for $G > 2$), there are few options available. In a method known as *one-against-all*, G SVM classifiers are obtained for each class separately and a new observation is assigned to the class chosen by maximum number of these classifiers (Bottou et al. 1994). In another method known as *one-against-one* (Kressel 1998), $G(G-1)/2$ SVM classifiers each separating a pair of classes are obtained and a new observation is assigned to the class that is predicted by the most classifiers (Kressel 1998). Hsu and Lin (2002) who compared their performances concluded that *one-against-one* performs better

than *one-against-all* in most of the situations that they studied. There are many modifications of SVM proposed by researchers from different fields that work better in certain specific situations. Typically, SVM works really well if there exists a good separation between classes or when the number of explanatory variables is large compared to the sample size of the training dataset. SVM is not computationally effective when using a very large training dataset.

3.4 Classification Trees

Classification trees (CT) are methods used to partition the space of explanatory variables into disjoint subsets and assign a class to each subset by minimizing some measure of misclassification also known as impurity. It is a visually pleasing method and can be easily as well as effectively described to those from the non-scientific communities. CT can handle large datasets as well as missing data, and it can easily ignore bad explanatory variables. However, sometimes depending on the dataset CT can produce a really bad partition of the space of explanatory variables leading to high misclassification rates.

CT produces a flowchart or tree structure starting with a root node (one explanatory variable) and then, proceeds with splits (internal nodes) until no split is deemed necessary (leaf nodes). Each leaf node is assigned to a class. There are many algorithms on how to select a root node, how to split a node, how many splits of each node are needed, and when to stop splitting a node to make it a leaf node. An example of classification tree is shown in Fig. 3. It shows classification of a random sample of $n = 78$ from iris data by R.A. Fisher (Anderson 1935) using JMP 12 into one of the three classes using two explanatory variables petal-width and petal-length.

The root node is typically chosen with an explanatory variable that provides the lowest rate of misclassification. This is easily achieved when the number of explanatory variables is small. For example, let there be two classes ($G = 2$) and one explanatory variable ($p = 1$). Consider the rule for using two complementary subgroups A_g created by a split with the explanatory variable such that, $y_i = g$ if $x_{1i} \in A_g$, $g = 1, 2$. For each split a Gini impurity measure is computed as,

$$I(CT) = \sum_{g=0}^{1} \left(1 - \sum_{j=0}^{1} \widehat{q_g(j)} \right)$$

where

$$\widehat{q_g(j)} = \frac{\sum_{i=1}^{n} I\left(y_i = j; x_{1i} \in A_g\right)}{\sum_{i=1}^{n} I\left(x_{1i} \in A_g\right)}$$

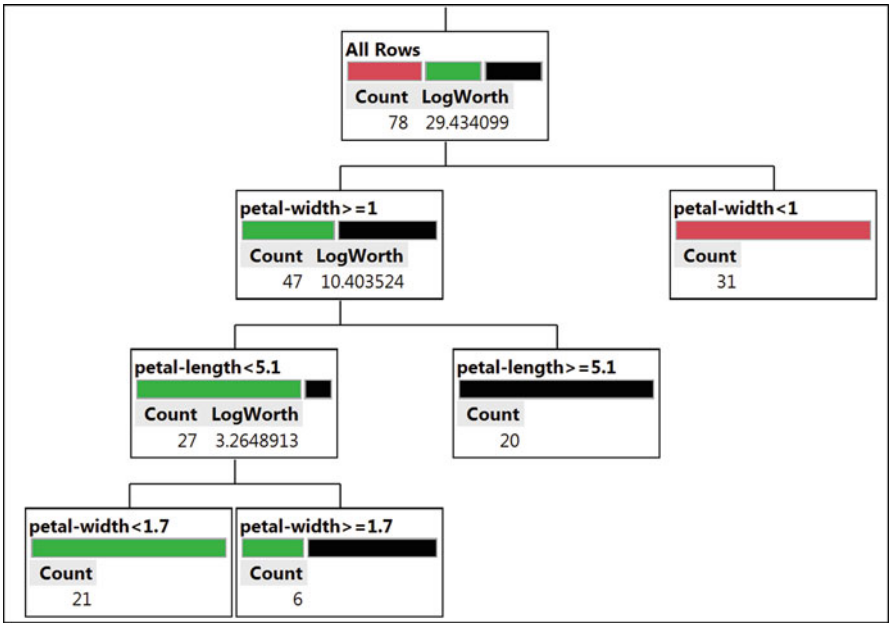

Fig. 3 Classification tree using a random sample of iris data ($n = 78$)

is the misclassification rate for group g and the splits are chosen such that the Gini impurity measure is minimized (Witten et al. 2011). As shown in Fig. 3, LogWorth for each model defined as $-\log_{10}(p\text{-value})$ is also another measure used to decide where to split. The p-value can be based on a chi-square test for a split that maximizes LogWorth value.

The first classification tree algorithm was proposed by Messenger and Mandell (1972). However, Breiman et al. (1984) have provided what became the most popular classification tree algorithm, namely classification and regression trees (CART). Several improved versions of it were proposed later and are still used in practice. Many modifications of the CART method have been proposed for various reasons, but mainly because CART produces biased and high-variance trees, i.e., changing the training set can drastically change the tree diagram. A few Bayesian versions of this algorithm are also available in the literature (Chipman et al. 1998; Denison et al. 1998). Loh (2009) provides a classification algorithm, called GUIDE which is computationally faster and incorporates nearest neighbor algorithm to improve the CT.

To reduce the variance in CT, two new methods were proposed, namely the bagging method (Breiman 1996) and the random forests method (Breiman 2001). Note that these methods do not produce a tree diagram but they focus on obtaining many classification trees from the training data so that a new observation that needs to be classified is put into the class suggested by majority of these trees. Bagging is simply achieved by obtaining bootstrap samples with replacement from the training

data. Random forests are similar to bagging but in each tree only a randomly chosen subset of typically $\sqrt{p}$ number of explanatory variables is considered when determining the nodes.

Freund and Schapire (1997) introduced the concept of boosting which aims to reduce bias in a CT. This is achieved by refitting the data into trees with higher weights for misclassified data points. In the initial calculation of the first CT, all data points are given equal weight. However, the weights are updated after each iteration and the impurity measure is updated by assigning higher weights to the misclassified observations. Then the final classifier is selected via weighted average of the trees.

4 Assessment and Comparison of the Performance of Classifiers

The performance of a classifier is typically judged by cross-validating the classification rule with a separate dataset of size s, called the validation data. Sometimes cross-validation is also used to estimate unknown parameters such as the number of neighbors to be considered in KNN method. In the absence of a separate validation data, the idea of Jackknife sampling (Quenouille 1949, 1956; Tukey 1958) is used to obtain a K-fold cross-validation. In this special case of cross-validation, the training dataset is divided into K smaller datasets of equal size, and $(K-1)$ of them are used as the training data and the remaining K^{th} one as the validation dataset. This process is repeated K times until each of them is used once as the validation data.

The simplest performance measure of a classifier is the misclassification rate $R(0 \leq R \leq 1)$, which is the proportion of validation sample that is misclassified. Hence a small value of $R(R \to 0)$ is an indication of more accurate classification. Although a very simple metric, this is an effective measure of performance. It works well as long as the cost of misclassification for and sample sizes from all classes are relatively similar. If sample sizes differ considerably, then the use of an uncertainty coefficient is recommended (Mills 2011). Uncertainty coefficient U is calculated as $U = (H - H_c)/H$ $(0 \leq U \leq 1)$ where

$$H = -\sum_{g=1}^{G} P\left(Y = g\right) \log\left(P\left(Y = g\right)\right)$$

and

$$H_c = -\sum_{g=1}^{G}\sum_{l=1}^{G} P\left(Y = g, \hat{Y} = l\right) \log\left(P\left(Y = g|\hat{Y} = l\right)\right)$$

can be estimated from the training data. Larger values of U indicate a better classifier. If the accuracy of classification for only one class is very important, then one can calculate the sensitivity (also known in medicine as the true positive rate or in machine learning as the recall rate) for that class. Sensitivity also takes values between 0 and 1 but a good classifier is expected to have a higher sensitivity. The sensitivity for class g can be calculated as,

$$sen_g = \frac{\sum_{i=1}^{n} I\left(y_i = g, \widehat{y_i} = g\right)}{\sum_{i=1}^{n} I\left(y_i = g, \widehat{y_i} = g\right) + \sum_{i=1}^{n} I\left(y_i = g, \widehat{y_i} \neq g\right)}$$

where I is the indicator variable taking values 1 or 0 depending on whether the condition is satisfied or not.

Besides misclassification rate, sensitivity, and uncertainty coefficient, there are many other performance measures available to judge classification methods, a detailed discussion of which is provided by Hand (2012).

Some articles dedicated to comparison of different classification methods are available. The earlier research was mostly focused on comparing logistic regression against LDA (Hosmer et al. 1983 and McLachlan and Byth 1979). Their outcomes indicate that LDA is a better performer if the explanatory variables are normally distributed but the advantage diminishes as sample size becomes larger. Meshbane and Morris (1996) recommended that QDA should be used instead of LDA if the distributions of explanatory variables are skewed. After comparing outcomes using classification tree and KNN, Liu and White (1995) concluded that KNN performs better than classification tree unless the number of explanatory variables is large. A study by Bhattacharya et al. (2011) compared SVM to logistic regression for detecting credit card fraud and found no advantage in using more complicated method like SVM over simpler logistic regression. Finch and Schneider (2006) compared performances of logistic regression, discriminant analysis, and classification trees based on simulated data while Kiang (2003) compared performances of logistic regression, LDA, KNN, and classification tree based on a separate set of simulated data. Asparoukhov and Krzanowski (2001) compared performances of all but one (namely SVM) classifiers mentioned in this paper using five real-life datasets for binary classification. They also discussed the effect of choosing different sized training set along with changing the number of explanatory variables. Steel et al. (2000) argue that simply comparing the methods is not completely meaningful unless the model selection process (i.e., the choice of explanatory variables in the final prediction model) is included. The only conclusion that can be drawn from all these studies is that there is no winner among all methods that work in every situation very effectively. The performance of a classifier depends very much on the dataset for which a classification is needed.

In this section, we compare the performance of six classifiers discussed earlier using two simulated datasets and three real-life datasets with respect to the misclassification rates and uncertainty measure U. All computations were done using available R packages *rpart*, *e1071*, *class*, *naivebayes*, and *MASS*. Of the two simulated datasets, one is visually separable albeit not linearly while the other is

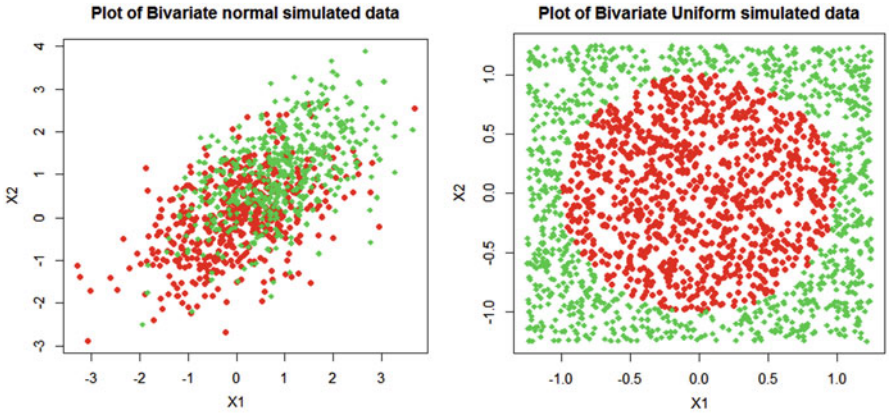

Fig. 4 Visualization of the simulated datasets

Table 1 Comparison of misclassification rates for different classifiers for five datasets

	Iris	Skin	Glass	Bivariate normal	Bivariate uniform
Logistic	0.09804	0.0401	NA	0.288	0.495
LDA	0.03922	0.0505	0.2364	0.300	0.495
NB	0.03922	0.0245	0.4545	0.304	0.077
KNN	0.03922	0.0048	0.1818	0.002	0.000
SVM	0.05882	0.0051	0.3273	0.312	0.008
CT	0.07843	0.0232	0.2545	0.320	0.096

Table 2 Comparison of uncertainty measure for different classifiers for five datasets

	Iris	Skin	Glass	Bivariate normal	Bivariate uniform
Logistic	0.71463	0.72119	NA	0.12654	0.00116
LDA	0.86424	0.71310	0.52570	0.11903	0.00116
NB	0.88589	0.80570	0.43788	0.11408	0.66483
KNN	0.88589	0.94258	0.62021	0.96238	1.00000
SVM	0.79531	0.93916	0.25449	0.06982	0.93602
CT	0.78020	0.80141	0.52453	0.10053	0.54546

not separable using any reasonable curve and provides a challenge in terms of classification (see Fig. 4). Both datasets have two explanatory variables as presented in scatterplots in Fig. 4 where each class is represented by separate point type and color. For NB classifier, Gaussian prior was used. For KNN classifier, the next larger odd integer to $\sqrt{n}$ was used as the value of K, except in one real data example (skin data), where this value was too large due to large dataset. The observed misclassification rates for six methods and five examples are listed in Table 1 and the uncertainty coefficients are listed in Table 2.

Example 1 (Iris) Consider the famous iris dataset by R.A. Fisher (Anderson 1935) described in the Introduction. Fifty observations are available for each type of iris. Of the 150 measurements available, a training dataset of 99 observations was created with 33 observations each from three groups. The misclassification rate was estimated based on the remaining 51 observations that constituted a validation sample. Misclassification rates lower than 0.10 (Table 1) and uncertainty measures over 0.70 (Table 2) show that all the methods did a commendable job of correct classification for this data. However, NB, KNN, and LDA are slightly better than other classifiers.

Example 2 (Skin) Refer to the skin segmentation dataset from the UCI machine learning repository (Bhatt et al. 2009). This dataset contains 245,057 observations randomly sampled from photos of faces of people of different age group, gender, and color. Of those, 50,589 observations are for samples of skin while the rest are for samples of non-skin parts of the face. The three explanatory variables in this example are RGB triplet, i.e., red, green, and blue colors used in displaying images. RGB values are typically given as an integer value in the range of 0–255, and combined together they determine the color of the image which in this case is part sampled. Distribution of RGB pixels for skin data is presented in a 3-dimensional plot in Fig. 5. Without rotating the plot around three axes it is difficult to tell if there is clear distinction or some overlap between two groups, skin and non-skin. A training sample of size 150,000 was used, out of which 30,000 were skin samples.

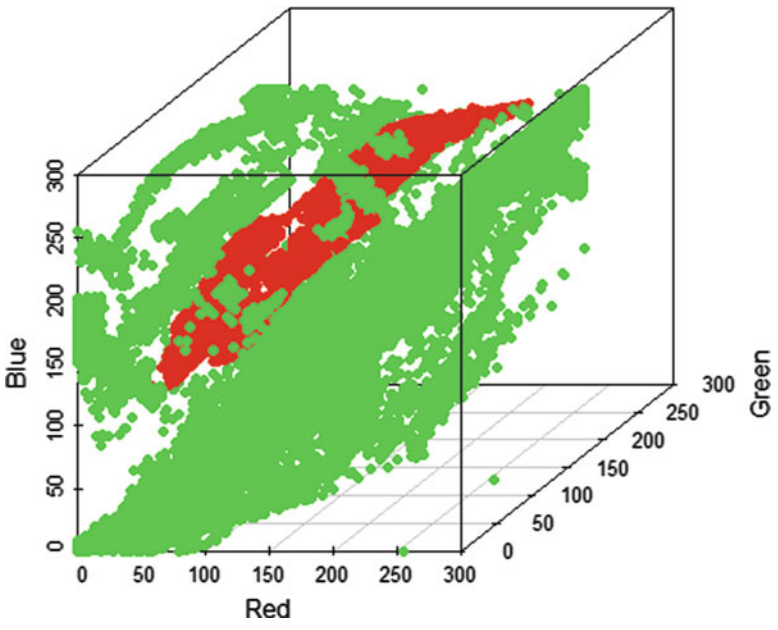

Fig. 5 Visualization of RGB pixel distribution for skin data

Tables 1 and 2 show that KNN and SVM perform best for this data, followed by NB and CT. To save computation time, $K = 19$ was used for KNN algorithm.

Example 3 (Glass) Consider the glass dataset from UCI machine learning repository (Lichman 2013). The original dataset describes six types of glass samples along with the refractive index and weight percent of oxides formed with sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron in the sample. Since some of the classes have small sample sizes, only three types of glass were used for classification purpose in this example. They are float-processed building window glasses (70 measurements), nonfloat-processed building window glasses (76 measurements), and non-window headlamp glasses (29 measurements). Fifty samples each from two classes of building window glasses and 20 samples from headlamp were used as the training data. Simulations to obtain parameters for a multinomial logistic regression failed due to non-convergence of iterations. Outcomes in Tables 1 and 2 indicate that KNN performs the best followed by CT and LDA.

Example 4 (Bivariate Uniform) Consider two independent univariate uniform distributions, namely $X_i \sim \mathit{Uniform}(-1.25, 1.25)$ for $i = 1, 2$. A sample of 3000 observations was generated with seed 1234. With the unit circle providing the class boundary, the i-th observation is assigned to group 1 if $x_{1i}^2 + x_{2i}^2 \leq 1$ and to group 2 otherwise. The first 2000 observations generated were used as a training data and the remaining 1000 as a validation sample. In this training dataset, 1012 observations were from group 1 and the remaining 998 from group 2. In the validation dataset, 520 observations were from group 1 and the remaining 480 from group 2. Note that in this situation a linear classifier is not supposed to perform well because of non-normal distributions and that is reflected in the misclassification rates listed in Table 1 and uncertainty measures listed in Table 2. Logistic regression and LDA seems to be only as good as a coin toss in this situation whereas KNN and SVM perform admirably well.

Example 5 (Bivariate Normal) Now consider the bivariate normal populations. A sample of 1500 observations from two homoscedastic bivariate normal distributions that differ only in mean vector was generated using the *mvtnorm* package in R with seed 5678, resulting in a total sample of size 3000. The difference in the mean vectors and the variance–covariance matrix used in the simulation were, respectively,

$$\mu_1 - \mu_2 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix}.$$

The first 1000 rows were used as a training sample for both groups (resulting in a total sample size of 2000) and the remaining 500 as the validation sample (resulting total sample size 1000). Tables 1 and 2 show KNN as a clear winner while the other classifiers are almost equally bad. Although we expected LDA to perform better, to our surprise the results say otherwise.

5 Concluding Remarks

In this paper, the basic ideas that dominate the world of statistical classification were described. Detailed discussions of them are scattered in different textbooks, but none discusses them all together. For example, logistic regression is discussed in detail by Kleinbaum and Klein (2010), LDA by McLachlan (2004), SVM by Steinwart and Christmann (2008), classification trees by Breiman et al. (1984), and different classification methods by Izenman (2008) and James et al. (2013).

For data with highly correlated explanatory variables or a large number of explanatory variables, the use of some dimension reduction technique such as principal component analysis, low variance filter, and high correlation filter before classification is recommended (Farcomeni and Greco 2015). In cases where $p > n$, dimension reduction becomes necessary. Alternatively, although random forests method is not a dimension reduction technique for explanatory variables, in cases where $\sqrt{p} < n$ this method can be effectively used without reducing dimension of explanatory variables.

A very basic question on this topic should be about the preference for any particular classification method. Alternatively, should there be preference for a certain classification method over the others. It depends on the circumstances. There is no single method that stands out as the best. Typically for complex problems in which the misclassification rate is higher among all classifiers, the use of soft classifiers is recommended. However, hard classifiers remain popular as their outcomes are easier to interpret in practice. Also hard classifiers like SVM and KNN generally provide good outcomes as seen from the situations discussed here. In this age of computation, the most recent research emphasis is on effective ways of implementing bagging and random forests (James et al. 2013) which can be computationally more effective than other classifiers like KNN. Liu et al. (2011) describe a suave large-margin unified machine that combines margin-based hard and soft classifiers, and that hard classifiers tend to perform better than soft classifiers when the classes are either easily separable or when the training sample size is relatively small compared to number of explanatory variables.

Research over the years has led to the development of many classifiers. As a result, the toolbox from which a classifier can be chosen provides an extensive list of options which to some extent depends on software used by and the computing power available for the researcher. Also comparative performance of different classifiers is changing with changing technology and results of past studies might lead to different conclusions with the current technology. Thus, one can entertain the idea of using all possible classifiers and assign a new observation to a class assigned by most of the classifiers.

References

Agresti, A.: Categorical Data Analysis. Wiley, Hoboken (2013)

Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: Theoretical foundations of the potential function method in pattern recognition learning. Autom. Remote. Control. **25**, 821–837 (1964)

Anderson, E.: The irises of the Gaspe Peninsula. Bull. Am. Iris Soc. **59**, 2–5 (1935)

Asparoukhov, O.K., Krzanowski, W.J.: A comparison of discriminant procedures for binary variables. Comput. Stat. Data Anal. **38**, 139–160 (2001)

Bayes, T.: An essay towards solving a problem in the doctrine of chances. Philos. Trans. **53**, 370–418 (1763)

Berkson, J.: Applications of the logistic function to bioassay. J. Am. Stat. Assoc. **9**, 357–365 (1944)

Bhatt, R.B., Sharma, G., Dhall, A., Chaudhury, S.: Efficient Skin Region Segmentation Using Low Complexity Fuzzy Decision Tree Model. IEEE-Indicon, Ahmedabad (2009)

Bhattacharya, S., Sanjeev, J., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: a comparative study. Decis. Support. Syst. **50**, 602–613 (2011)

Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory – COLT '92. p. 144 (1992)

Bottou, L., Cortes, C., Denker, J.S., Drucker, L., Guyon, I., Jackel, L., LeCun, Y., Muller, U.A., Sackinger, E., Simard, P., Vapnik, V.N.: Comparison of classifier methods: a case study in handwriting digit recognition. Int. Conf. Pattern Recognit. **2**, 77–87 (1994)

Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)

Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth, Belmont (1984)

Burrus, C.S., Barreto, J.A., Selesnick, I.W.: Iterative reweighted least squares design of FIR filters. IEEE Trans. Signal Process. **42**(11), 2922–2936 (1994)

Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian CART model search. J. Am. Stat. Assoc. **93**, 935–948 (1998)

Chomboon, K., Pasapichi, C., Pongsakorn, T., Kerdprasop, K., Kerdprasop, N.: An empirical study of distance mreics for K-nearest neighbor algorithm. 3rd International Conference on Industrial Application Engineering, 280–285 (2015)

Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)

Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. **13**(1), 21–27 (1967)

Cox, D.R.: Analysis of Binary Data. Chapman and Hall, London (1969)

Cramer, J.S.: The origins and development of the logit model. In: Cramer, J.S. (ed.) Logit Models from Economics and Other Fields, pp. 149–158. Cambridge University Press, Cambridge (2003)

Denison, D.G.T., Mallick, B.K., Smith, A.F.M.: A Bayesian CART algorithm. Biometrika. **85**, 363–377 (1998)

Farcomeni, A., Greco, L.: Robust Methods for Data Reduction. CRC Press, Boca Raton (2015)

Finch, W.H., Schneider, M.K.: Misclassification rates for four methods of group classification. Educ. Psychol. Meas. **66**(2), 240–257 (2006)

Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugenics. **7**(2), 179–188 (1936)

Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **5**(1), 119–139 (1997)

Grabusts, P.: The choice of metrics for clustering algorithms. Proceedinhs of the 8th International Scientific and Practical Conference, **11**, 70–76 (2011)

Gurland, J., Lee, I., Dahm, P.A.: Polychotomous quantal response in biological assay. Biometrics. **16**, 382–398 (1960)

Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950)
Hand, D.J.: Classifier technology and the illusion of progress. Stat. Sci. **21**, 1–14 (2006)
Hand, D.J.: Assessing the performance of classification methods. Int. Stat. Rev. **80**, 400–414 (2012)
Hand, D.J., Yu, K.: Idiot's Bayes - not so stupid after all? Int. Stat. Rev. **69**(3), 385–399 (2001)
Har-Peled, S., Indyk, P., Motwani, R.: Approximate nearest neighbor: towards removing the curse of dimensionality. Theory Comput. **8**, 321–350 (2012)
Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, New York (2001)
Hosmer, T., Hosmer, D.W., Fisher, L.L.: A comparison of the maximum likelihood and discriminant function estimators of the coefficients of the logistic regression model for mixed continuous and discrete variables. Commun. Stat. **12**, 577–593 (1983)
Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)
Izenman, A.J.: Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning. Springer, New York (2008)
James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning: With Applications in R. Springer, New York (2013)
Kiang, M.: A comparative assessment of classification methods. Decis. Support. Syst. **35**, 441–454 (2003)
Kleinbaum, D.G., Klein, M.: Logistic Regression: A Self-learning Text, 3rd edn. Springer, New York (2010)
Kressel, U.H.G.: Pairwise classification and support vector machines. In: Advances in Kernel Methods: Support Vector Learning, pp. 255–268. MIT Press, Cambridge (1998)
Lange, K.: MM Optimization Algorithms. SIAM, Philadelphia (2016)
Lichman, M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine 2013
Liu, W.Z., White, A.P.: A comparison of nearest neighbor and tree-based methods of non-parametric discriminant analysis. J. Stat. Comput. Simul. **53**, 41–50 (1995)
Liu, Y., Zhang, H.H., Wu, Y.: Hard or soft classification? Large-margin unified machines. J. Am. Stat. Assoc. **106**(493), 166–177 (2011)
Loh, W.Y.: Improving the precision of classification trees. Ann. Appl. Stat. **3**, 1710–1737 (2009)
Mahalanobis, P.C.: On the generalized distance in statistics. Proceedings of the National Institute of Science in India, **2**(1), 49–55, (1936)
Mai, Q., Zou, H.: Semiparametric sparse discriminant analysis in ultra-high dimensions. J. Multivar. Anal. **135**, 175–188 (2015)
Mantel, N.: Models for complex contingency tables and polychotomous response curves. Biometrics. **22**, 83–110 (1966)
McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. Wiley-Interscience, New York (2004)
McLachlan, G.J., Byth, K.: Expected error rates for logistic regression versus normal discriminant analysis. Biom. J. **21**, 47–56 (1979)
Menard, S.: Applied Logistic Regression Analysis, 2nd edn. Sage Publications, Thousand Oaks (2002)
Meshbane, A., Morris, J.D.: A method for selecting between linear and quadratic classification models in discriminant analysis. J. Exp. Educ. **63**(3), 263–273 (1996)
Messenger, R., Mandell, L.: A modal search technique for predictive nominal scale multivariate analysis. J. Am. Stat. Assoc. **67**, 768–772 (1972)
Mills, P.: Efficient statistical classification of satellite measurements. Int. J. Remote Sens. **32**, 6109–6132 (2011)
Mulekar, M.S., Brown, C.S.: Distance and similarity measures. In: Alhaji, R., Rekne, J. (eds.) Encyclopedia of Social Network and Mining (ESNAM), pp. 385–400. Springer, New York (2014)
Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)

Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Mach. Learn. **39**(2-3), 103–134 (2000)
Nilsson, N.: Learning Machines: Foundations of Trainable Pattern-Classifying Systems. McGraw-Hill, New York (1965)
Quenouille, M.H.: Problems in plane sampling. Ann. Math. Stat. **20**(3), 355–375 (1949)
Quenouille, M.H.: Notes on bias in estimation. Biometrika. **43**(3-4), 353–360 (1956)
Rao, R.C.: The utilization of multiple measurements in problems of biological classification. J. R. Stat. Soc. Ser. B. **10**(2), 159–203 (1948)
Rish, I.: An empirical study of the naive Bayes classifier. In: IJCAI Workshop on Empirical Methods in AI, Sicily, Italy (2001)
Samworth, R.J.: Optimal weighted nearest neighbour classifiers. Ann. Stat. **40**(5), 2733–2763 (2012)
Schölkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., Vapnik, V.: Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans. Signal Process. **45**, 2758–2765 (1997)
Sebestyen, G.S.: Decision-making Process in Pattern Recognition. McMillan, New York (1962)
Snell, E.J.: A scaling procedure for ordered categorical data. Biometrics. **20**, 592–607 (1964)
Soria, D., Garibaldi, J.M., Ambrogi, F., Biganzoli, E.M., Ellis, I.O.: A non-parametric version of the naive Bayes classifier. Knowl.-Based Syst. **24**(6), 775–784 (2011)
Srivastava, S., Gupta, M.R., Frigyik, B.A.: Bayesian quadratic discriminant analysis. J. Mach. Learn. Res. **8**, 1287–1314 (2007)
Steel, S.J., Louw, N., Leroux, N.J.: A comparison of the post selection error rate behavior of the normal and quadratic linear discriminant rules. J. Stat. Comput. Simul. **65**, 157–172 (2000)
Steinwart, I., Christmann, A.: Support Vector Machines. Springer, New York (2008)
Theil, H.: A multinomial extension of the linear logit model. Int. Econ. Rev. **10**(3), 251–259 (1969)
Tjalling, J.Y.: Historical development of the Newton-Raphson method. SIAM Rev. **37**(4), 531–551 (1995)
Tukey, J.W.: Bias and confidence in not quite large samples. Ann. Math. Stat. **29**(2), 614–623 (1958)
Vapnik, V., Lerner, A.: Pattern recognition using generalized portrait method. Autom. Remote. Control. **24**, 774–780 (1963)
Wahba, G.: Soft and hard classification by reproducing Kernel Hilbert space methods. Proc. Natl. Acad. Sci. **99**, 16524–16530 (2002)
Witten, I., Frank, E., Hall, M.: Data Mining. Morgan Kaufmann, Burlington (2011)