Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

ICEENG

6th International Conference
on Electrical Engineering
ICEENG 2008

Statistical learning machines from ATR to DNA micro arrays: design, assessment, and advice for practitioners

By

Waleed A. Yousef*

Abstract:

Statistical Learning is the process of estimating an unknown probabilistic input-output relationship of a system using a limited number of observations; and a statistical learning machine (SLM) is the machine that learned such a process. While their roots grow deeply in Probability Theory, SLMs are ubiquitous in the modern world. Automatic Target Recognition (ATR) in military applications, Computer Aided Diagnosis (CAD) in medical imaging, DNA microarrays in Genomics, Optical Character Recognition (OCR), Speech Recognition (SR), spam email filtering, stock market prediction, etc., are few examples and applications for SLM; diverse fields but one theory.

The field of Statistical Learning can be decomposed to two basic subfields, Design and Assessment. We mean by Design, choosing the appropriate method that learns from the data to construct an SLM that achieves a good performance. We mean by Assessment, attributing some performance measures to the designed SLM to assess this SLM objectively. To achieve these two objectives the field encompasses different other fields: Probability, Statistics and Matrix Theory; Optimization, Algorithms, and programming, among others.

Three main groups of specializations—namely statisticians, engineers, and computer scientists (ordered ascendingly by programming capabilities and descendingly by mathematical rigor)—exist on the venue of this field and each takes its elephant bite. Exaggerated rigorous analysis of statisticians sometimes deprives them from considering new ML techniques and methods that, yet, have no “complete” mathematical theory. On the other hand, immoderate add-hoc simulations of computer scientists sometimes derive them towards unjustified and immature results. A prudent

* Faculty of Computers and Information, Helwan University.

approach is needed that has the enough flexibility to utilize simulations and trials and errors without sacrificing any rigor. If this prudent attitude is necessary for this field it is necessary, as well, in other fields of Engineering.

In the spirit of this prelude, this article is intended to be a pilot-view of the field that sheds the light on SLM applications, the Design and Assessment stages, necessary mathematical and analytical tools, and some state-of-the-art references and research.

Keywords:

Statistical Learning, Machine Learning, Pattern Recognition, Pattern Classification, Automatic Target Recognition, Computer Aided Diagnosis, Classifier Assessment, Receiver Operating Characteristics

1. Introduction and Terminology

In the present section some basic concepts and terminology necessary for the sequel will be formally introduced. The world of variables can be categorized into two categories: deterministic variables and random variables. A deterministic variable takes a definite value; the same value will be the outcome if the experiment that yielded this value is rerun. On contrary, a random variable is a variable that takes a non-definite value with a probability value.

Definition 1:
A random variable X *is a function from a sample space* S *into the real numbers* R, *that associates a real number,* $x = X(s)$, *with each possible outcome* s ج S.

Details on the topic can be found in [1, Ch. 1]. For more rigorous treatment of random variables based on measure theoretic approach see [2]. Variables can be categorized as well, based on value, into: quantitative or metric, qualitative or categorical, and ordered categorical A quantitative variable takes a value on R and it can be discrete or continuous. A qualitative or categorical variable does not necessarily take a numerical value; rather it takes a value from a finite set. E.g., the set $G = \{\mathrm{Red}, \mathrm{Green}, \mathrm{Blue}\}$ is a set of possible qualitative values that can be assigned to a color. An ordered categorical variable is a categorical variable with relative algebraic relations among the values. E.g., the set $G = \{\mathrm{Small}, \mathrm{Medium}, \mathrm{Large}\}$ includes ordered categorical values.

Variables in a particular process are related to each other in a certain manner. When variables are random the process is said to be stochastic, i.e., when the inputs of this process have some specified values there is no deterministic value for the output, rather a probabilistic one. The output in this case is a random variable.

We next consider the general problem of statistical learning algorithms. Consider a sample consisting of a number of cases—the words cases and observations may be used exchangeably—, where each case is composed of the set of inputs that will be given to the algorithm together with the corresponding output. Such a sample provides the means for the algorithm to learn during its so-called “design” stage. The goal of this learning or design stage is to understand as much as possible how the output is related to the inputs in these observations, so that when a new set of inputs is given in the future the algorithm will have some means of predicting the corresponding output. The above terminology has been borrowed from the field of machine learning. This problem is originally from the field of statistical decision theory, where the terminology is somewhat different. In the latter field, the inputs are called the predictors and the output is called the response. When the output is quantitative the learning algorithm is called regression; when the output is categorical or ordered categorical the learning algorithm is called classification. In the engineering communities that work on the pattern classification problem, the terms input features and output class are used respectively. The learning process in that setting is called training and the algorithm is called the classifier.

Definition 2:

Learning is the process of estimating an unknown input-output dependency or structure of a system using a limited number of observations.

Statistical learning is crucial to many applications. One of the first applications that utilized learning was Automatic Target Recognition (ATR) in military applications. In the medical imaging field, a tumor on a mammogram must be classified as malignant or benign. This is an example of prediction, regardless of whether it is done by a radiologist or by a computer algorithm (Computer Aided Diagnosis or CAD). In either case the prediction is done based on learning from previous mammograms. The features, i.e., predictors, in this case may be the size of the tumor, its density, various shape parameters, etc. The output, i.e., response, is a categorical one which belongs to the set $G = \{\text{benign}, \text{malignant}\}$. There are so many such examples in biology and medicine that it is almost a field unto itself, i.e., biostatistics. The task may be diagnostic as in the mammographic example, or prognostic where, for example, one estimates the probability of occurrence of a second heart attack for a particular patient who has had a previous one. All of these examples involve a prediction step based on previous learning. A wide range of commercial and military applications arises in the field of satellite imaging. Predictors in this case can be measures from the image spectrum, while the response can be the type of land or crop or vegetation of which the image was taken.

The biomedical landmark of our time has been the sequencing of the genomes of many organisms, in particular, the sequencing of the human genome. The availability of this version of life's instruction book is leading to a very great horizon of research possibilities, including many approaches to personalized medicine. In Genome project the interest is making a prediction of the number of genes in the human genome. One of the key tools for this task is a family of learning machines referred to as "Hidden Markov Models" (HMMs). On the other hand, in Genomics, the task is to predict the level of expression of a particular gene responsible for a particular disease. The set of predictors in this case are multiple-gene-expression microarrays ("DNA chips").

Before going through some mathematical details, it is convenient to introduce some commonly used notation. A random variable—or a random vector—is referred to by an upper-case letter, e.g., X. An instance, or observation, of that variable is referred to by a lower-case letter, e.g., x. A collection of N observations for the p-dimensional random vector X is collected into an $N \times P$ matrix and represented by a bold upper-case $\mathbf{X}$. A lower-case bold letter $\mathbf{x}$ is reserved for describing a vector of any N-observations of a variable, even a tuple consisting of non-homogeneous types. The main notation in the sequel will be as follows: $\mathbf{t} : \{t_i = (x_i, y_i)\}$ represents an n-case training data set, i.e., one on which the learning mechanism will execute. Every sample case t_i of this set represents a tuple of the predictors x_i represented in a p-dimensional vector, and the corresponding response variable y_i. All the N observations x_i's may be written in a single $N \times P$ matrix $\mathbf{X}$, while all the observations y_i may be written in a vector $\mathbf{y}$.

1. Statistical Decision Theory

This section provides an introduction to statistical decision theory, which serves as the foundation of statistical learning. If a random vector X and a random variable Y have a joint probability density $f_{X,Y}(x,y)$, the problem is defined as follows: how to predict the variable Y from an observed value for the variable X. In this section we assume having a full knowledge of the joint density $f_{X,Y}$, so there is no learning yet (Definition 1). The prediction function $h(X)$ is required to have minimum average prediction error. The prediction error should be defined in terms of some loss function $L(Y, h(X))$ that penalizes for any deviation in the predicted value of the response from the correct value. Define the predicted value by:

$$\hat{Y} = h(X) \tag{1}$$

The risk of this prediction function is defined by the average loss, according to the defined loss function, for the case of prediction:

$$R(h) = E\left[L(Y, \hat{Y})\right] \tag{2}$$

For instance, some constraint will be imposed on the response Y by assuming it, e.g., to be a quantitative variable. This is the starting point of the statistical branch of regression, where (1) is the regression function. A form should be assumed for the loss function. A mathematically convenient and widely used form is the squared-error loss function:

$$L(Y, h(X)) = (Y - h(X))^2 \tag{3}$$

In this case (2) becomes:

$$R(h) = \int (Y - h(X))^2 \, dF_{X,Y}(X,Y) \tag{4}$$

$$= E_X\left[E_{Y|X}\left[(Y - h(X))^2 \mid X\right]\right] \tag{5}$$

Hence, (5) is minimized by minimizing the inner expectation over every possible value for the variable X. Ordinary vector calculus solves the minimization for $h(X)$ and gives:

$$h(X) = \arg\min_{h(X)} \left(E_{Y|X} \left[(Y - h(X))^2 \mid X \right] \right) \tag{6}$$

$$= E_Y [Y \mid X] \tag{7}$$

(A more common proof that does not require regularity conditions, which assume differentiability under the integration sign, can be found in [3]). This means that if the joint distribution for the response and predictor is known, the best regression function in the sense of minimizing the risk is the expectation of the response conditional on the predictor. In that case the risk of regression in (5) will be:

$$R_{\min}(h) = E_X [Var [Y \mid X]]$$

Recalling (2), and lifting the constraint on the response being quantitative, and setting another constraint by assuming it to be a qualitative (or categorical) variable gives rise to the classification problem. Now the loss function cannot be the squared-error loss function defined in (3), since this has no meaning for categorical variables. Since Y may take now a qualitative value from a set of size k, (see Section 0), the loss function can be defined by the matrix

$$L(Y, h(X)) = ((c_{ij})), \quad 1 < i, \quad j < k \tag{8}$$

where the non-negative element c_{ij} is the cost, the penalty or the price, paid for classifying an observation as y_j when it belongs to y_i. In the field of medical decision making this is often called the *utility matrix*. Under this assumption, the risk defined by (2) can be rewritten for the categorical variables to be:

$$R(h) = E_X E_{Y|X} [L(Y, h(X))] \tag{9}$$

$$= E_X \left[\sum_{i=1}^{k} c_{ij} \Pr[Y = y_i \mid X] \right], \tag{10}$$

where $\Pr[Y \mid X]$ is the probability mass function for Y conditional on X. Then the conditional risk for decision y_j

$$R(j, h) = \sum_{i=1}^{k} c_{ij} \Pr[Y = y_i \mid X] \tag{11}$$

is the expected loss when classifying an observation as belonging to y_j and the expectation is taken over all the possible values of the response. Again, (10) can be minimized by minimizing the inner expectation to give:

$$h(X) = \arg\min_{j}\left[\sum_{i=1}^{k} c_{ij}\Pr[Y = y_i \mid X]\right] \tag{12}$$

Expressing the conditional probability of the response in terms of Bayes law and substituting in (12) gives:

$$h(X) = \arg\min_{j}\sum_{i=1}^{k} c_{ij} f_X(X \mid Y = y_i)\Pr[y_i] \tag{13}$$

Where $\Pr[y_i]$ is the prior probability for y_j while $\Pr[y_j \mid X]$ is the posterior probability; i.e., the probability that the observed case belongs to y_j, given the value of X. This is what statisticians call Bayes classification, or Bayes decision rule or alternatively, what engineers call the Bayes classifier.

Some special cases here may be of interest. The first case is when equal costs are assigned to all misclassifications and there is no cost for correct classification; this is called the 0-1 cost function. This reduces (12) to:

$$h(X) = \arg\min_{j}[1 - \Pr[Y = y_j \mid X]] \tag{14}$$

$$= \arg\max_{j}[\Pr[Y = y_j \mid X]] \tag{15}$$

The rule thus is to classify the sample case to the class having maximum posterior probability. Another special case of great interest is binary classification, i.e., the case of $k = 2$. In this case (12) reduces to:

$$\frac{\Pr[y_1 \mid X]}{\Pr[y_2 \mid X]} \underset{y_2}{\overset{y_1}{\gtrless}} \frac{(c_{22} - c_{21})}{(c_{11} - c_{12})} \tag{16}$$

Alternatively, this can be expressed as :

$$\frac{f_X(X = x \mid y_1)}{f_X(X = x \mid y_2)} \underset{y_2}{\overset{y_1}{\gtrless}} \frac{\Pr[y_2](c_{22} - c_{21})}{\Pr[y_1](c_{11} - c_{12})} \tag{17}$$

The decision taken in (12) has the minimum risk, which can be calculated by substituting back in (10) to give:

$$R_{\min}(h) = \sum_{i=1}^{k}\int_{X} c_{i,j(X)}\Pr[y_i]\,dF_X(X \mid y_i) \tag{18}$$

where $j(X)$ is the class decision $h(X)$. For binary classification and where there is no cost for a correct decision, i.e., $c_{11} = c_{22} = 0$, this reduces to:

$$R_{\min}(h) = c_{12}\Pr[y_1]\int_{R_2} dF_X(X \mid y_1) + c_{21}\Pr[y_2]\int_{R_1} dF_X(X \mid y_2) \tag{19}$$

where each of R_1 and R_2 is the predictor hyperspace over which the optimum decision (16) predicts as class 1 or class 2 respectively. Latter, the response variable Y may be referred to W in case of classification. To follow the notation of Section 0 the response of an observation is assigned a value $w_i, i = 1,\ldots,k$ to express a certain class.

To recap, this section emphasizes the fact that there is no distinction between regression and classification from the conceptual point of view. Each minimizes the risk of predicting the response variable for an observation, i.e., a sample case with known predictor(s). If the joint probability distribution function for the response and predictors is known, it is just a matter of direct substitution in the above results. If the joint distribution is known but its parameters are not known, a learning process is used to estimate those parameters from a training sample $\mathbf{t}$ by methods of statistical inference; see [4], and [1]. However, if the joint distribution is unknown, this gives rise to two different branches of prediction. These two branches are parametric regression (or classification), introduced in Section 1—where the regression or classification function is modeled and a training sample is used to build that model—and nonparametric regression (or classification), introduced in Section 2—where no particular parametric model is assumed.

1. Parametric Regression and Classification

The prediction method introduced in Section 1 assumes, as indicated, that the joint density of the response and the predictor is known. If such knowledge exists, all the methods revolve around modeling the regression function (1) in the case of regression or the posterior probabilities in (12) in the case of classification.

1.1. Linear Models

In linear model (LM) theory, Y is assumed to be in the form:

$$Y = E[Y] + e \tag{20}$$

$$= a + X^{¢}b + e \tag{21}$$

where the randomness of Y comes only from e, and it is assumed that the conditional expectation of Y is linear in the predictors X. The two basic assumptions in the theory are the zero mean and constant variance of the random error component e. The regression function (1) is then written as:

$$h(X) = a + X^{¢}b \tag{22}$$

More generally, still a linear model, it can be rewritten as:

$$h(X) = X^{¢}_{new}b, \tag{23}$$

$$X^{¢}_{new} = (f_1(X),\ldots,f_d(X)) \tag{24}$$

where the predictor X is replaced by a new d-dimensional vector, X_{new}, whose elements are scalar functions of the random vector X.

The intercept a in (22) may be modeled, if needed, in terms of (23) by setting

$f_1(X) = 1$. Equation (23) can be seen as equivalent to (22), where X has been transformed to X_{new} which became the new predictor on which Y will be regressed.

Now β must be estimated, and this point estimation is done for some observed values of the predictor. Writing the equations for n observed values gives:

$$\mathbf{y} = \mathbf{X}'\beta + \boldsymbol{\varepsilon} \qquad (25)$$

If (25) is solved for β to give the least sum of squares for the components of error vector $\boldsymbol{\varepsilon}$, this will give, as expected, the same result as if we approximated the conditional expectation of Y by the set of observations $\mathbf{y}$. Solving either way gives:

$$\hat{\beta} = \left(\mathbf{X}\mathbf{X}'\right)^{-1}\mathbf{X}\mathbf{y} \qquad (26)$$

Then the prediction of Y is done by estimating its expectation which is given by:

$$\hat{h}(X) = \hat{E}[Y] = X'\hat{\beta} \qquad (27)$$

For short notation we always write $\hat{Y}$ instead of $\hat{E}[Y]$.

Nothing up to this point involves statistical inference. This is just fitting a mathematical model using the squared-error loss function. Statistical inference starts when considering the random error vector $\boldsymbol{\varepsilon}$ and the effect of that on the confidence interval for $\hat{\beta}$ and the confidence in predicted values of the response for particular predictor variable, or any other needed inference. All of these important questions are answered by the theory of linear models. A very good reference for an applied approach to linear models, without any mathematical proofs, is [5]. For a theoretical approach and derivations the reader is referred to [6], [3], and [7]. For very rigorous mathematical treatment for the theory of testing statistical hypothesis the reader should visit [8]. It is remarkable that if the joint distribution for the response and the predictor is multinormal, the linear model assumption (21) is an exact expression for the random variable Y. This fact arises from the fact that the conditional expectation for the multinormal distribution is linear in the conditional variable. That is, by assuming that

$$\begin{pmatrix} Y \\ X \end{pmatrix} \sim N(\mu, \Sigma), \text{ where} \qquad (29)$$

$$\mu = \begin{pmatrix} \mu_Y \\ \mu_X \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \qquad (32)$$

then the conditional expectation of Y on X is given by:

$$E[Y \mid X = x] = \mu_Y + \Sigma_{12}\Sigma_{22}^{-1}(x - \mu_X) \qquad (33)$$

For more details on the multinormal properties, see [9].

In the case of classification the classes are categorical variables but a dummy variable can be used as coding for the class labels. Then a linear regression is carried out for this dummy variable on the predictors. A drawback of this approach is what is called class masking, i.e., if more than two classes are used, one or more can be masked by others and they may not be assigned to any of the observations in prediction. For a clear example of masking see [10, Sec. 4.2].

1.2. Generalized Linear Models

In linear models the response variable is directly related to the regression function by a linear expression of the form of (21). In many cases a model can be improved by indirectly relating the response to the predictor through a linear model—some times it is necessary as will be shown for the classification problem. This is done through a transformation or *link* function g by assuming:

$$g(E[Y]) = X'\beta \tag{34}$$

Now it is the transformed expectation that is modeled linearly. Hence, linear models are merely a special case of the generalized linear models when the link function is the identity function $g(E[Y]) = E[Y]$.

A very useful link function is the *logit* function defined by:

$$g(\mu) = \log\frac{\mu}{1-\mu}, \quad 0 < \mu < 1 \tag{35}$$

Through this function the regression function is modeled in terms of the predictor as:

$$E[Y] = \frac{\exp(X'\beta)}{1+\exp(X'\beta)} \tag{36}$$

which is known as logistic regression. Equation (36) implies a constraint on the response Y, i.e., it must satisfy $0 < E[Y] < 1$, a feature that makes logistic regression an ideal approach for modeling the posterior probabilities in (12) for the classification problem. Equation (35) models the two-class problem, i.e., binary classification, by considering the new responses Y_1 and Y_2 to be defined in terms of the old responses ω_1 and ω_2, the classes, as:

$$Y_1 = \Pr[\omega_1 \mid X], \tag{37}$$

$$Y_2 = \Pr[\omega_2 \mid X] = 1 - \Pr[\omega_1 \mid X] \tag{38}$$

The general case of the k-class problem can be modeled using $K-1$ equations, because of the constraint $\sum_i \Pr[\omega_i \mid X] = 1$, as:

$$\log\frac{\Pr[\omega_i \mid X = x]}{\Pr[\omega_k \mid X = x]} = x'\beta_i, \quad i = 1,\ldots,K-1 \tag{39}$$

Alternatively, (39) can be rewritten as:

$$\Pr[\omega_i \mid X = x] = \frac{\exp(x'\beta_i)}{1 + \sum_{j=1}^{K-1} \exp(x'\beta_j)}, \quad 1 \le i \le K-1, \tag{40}$$

$$\Pr[\omega_k \mid X = x] = \frac{1}{1 + \sum_{j=1}^{K-1} \exp(x'\beta_j)} \tag{41}$$

The question now is how to estimate $\beta_i \ \forall \ i$. The multinomial distribution for modeling observations is appropriate here. For illustration, consider the case of binary classification; the log-likelihood for the n-observations can then be written as:

$$l(\beta) = \sum_{i=1}^{n} \{y_i \log \Pr[\omega_1 \mid X_i, \beta] + (1 - y_i) \log(1 - \Pr[\omega_1 \mid X_i, \beta])\} \tag{42}$$

$$= \sum_{i=1}^{n} \left\{ y_i x_i'\beta - \log(1 + e^{x_i'\beta}) \right\} \tag{43}$$

To maximize this likelihood, the first derivative is set to zero to obtain:

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{n} x_i (y_i - \frac{e^{x_i'\beta}}{1 + e^{x_i'\beta}}) \text{set} = 0 \tag{44}$$

This is a set of k equations, where the vector X can be the original predictor $(x_1, \ldots, x_p)'$ or any transformation $(f_1(X), \ldots, f_d(X))'$ as in (24). Equation (44) is a set of non-linear equations, and can be solved by iterative numerical methods like the Newton-Raphson algorithm. For more details with numerical examples see [10, Sec. 4.4] or [1, Sec. 12.3].
It can be noted that (42) is valid under the assumption of the following general distribution:

$$f(X) = f(\theta_i, \gamma) h(X, \gamma) \exp(\theta_i' X) \tag{45}$$

with probability π_i, $i = 1, 2$, $\pi_1 + \pi_2 = 1$, which is the exponential family. So logistic regression is no longer an approximation for the posterior class probability if the distribution belongs to the exponential family. For insightful comparison between logistic regression and the Bayes classifier under the multinormal assumption see [11].

It is very important to mention that logistic regression, and all subsequent classification methods, assume equal a priori probabilities. Then the ratio between the posterior probabilities will be the same as the ratio between the densities that appear in (13). Hence, the estimated posterior probabilities from any classification method are used in (13) as if they are the estimated densities.

1.3. Non-linear Models

The link function in the generalized linear models is modeled linearly in the predictors, (34). Consequently, the response variable is modeled as a non-linear function. In contrast to the linear models described in Section 0, in non-linear models the response can be modeled non-linearly right from the beginning, without the need for a link function.

2. Nonparametric Regression and Classification

In contrast to parametric regression, the regression function (1) is not modeled parametrically, i.e., there is no particular parametric form to be imposed on the function. Nonparametric regression is a versatile and flexible method of exploring the relationship of two variables. It may appear that this technique is more efficient than the linear models, but this is not the case. Linear models and nonparametric models can be thought of as two different techniques in the analyst's toolbox. If there is an a priori reason to believe that the data follow a parametric form, then linear models or parametric regression in general may provide an argument for an optimal choice. If there is no prior knowledge about the parametric form the data may follow or no prior information about the physical phenomenon that generated the data, there may be no choice other than nonparametric regression.

There are many nonparametric techniques proposed in the statistical literature. Some of these techniques have also been developed in the engineering community under different names, e.g., artificial neural networks. What was said above, when comparing parametric and nonparametric methods, can also be said when comparing nonparametric methods to each other. None can be preferred overall across all situations.

This section introduces some of the nonparametric regression and classification methods. The purpose is not to present a survey as much as to introduce the topic and show how it relates with the parametric methods to serve one purpose, predicting a response variable, categorical or quantitative. An excellent comprehensive source for regression and classification methods, with practical approaches and illustrative examples, is [10].

2.1. Smoothing Techniques

Smoothing is a tool for summarizing in a nonparametric way a trend between a response and a predictor such that the resulting relationship is less variable than the original response, hence the name smoothing. When the predictor is unidimensional, the smoothing is called scatter-plot smoothing. In this section, some methods used in scatter-plot smoothing are considered. These smoothing methods do not succeed in higher dimensionality. This is one bad aspect of what is called the curse of dimensionality, which will be discussed in Section 5.

2.1.1. K ***-Nearest Neighbor***

The regression function (1) is estimated in the K -nearest neighbor approach by:

$$h(x) = \frac{1}{n}\sum_{i=1}^{n} W_i(x)y_i, \tag{46}$$

$$W_i(x) = \begin{cases} n:k & = i \quad J_x \equiv \{i / x_i \in N_k(x)\} \; (47) \\ 0 & \text{otherwise} \end{cases} \tag{48}$$

where $N_k(x)$ is the set consisting of the nearest k points to the point x. So in the case of regression, this technique approximates the conditional mean, i.e., the regression function that gives minimum risk, by local averaging for the response Y . In the case of classification, the posterior probability is estimated by:

$$\Pr[w_j \mid x] = \frac{1}{n}\sum_{i=1}^{n} W_i(x) I_{w_i = w_j}, \tag{49}$$

and I is the indicator function defined by:

$$I_{cond} = \begin{cases} 1 & \text{cond is True} \\ 0 & \text{cond is False} \end{cases}. \tag{50}$$

This accounts for replacing the continuous response in (46) by an indicator function for each class given each point. So, the posterior probability is approximated by a frequency of occurrence in a k -point neighborhood.

2.1.2. ***Nearest Neighbor***

This is a special case of the K -nearest neighbor method where $k = 1$. It can be thought of as narrowing the window W on which regression is carried out. In effect, this makes the regression function or the classifier more complex because it is trying to estimate the distribution at each point.

2.1.3. ***Kernel Smoothing***

In this approach a kernel smoothing function is assumed. This means that a weighting and convolution (or mathematical smoothing) is carried out for the points in the neighborhood of the predicted point according to the chosen kernel function. Formally this is expressed as:

$$h(x) = \sum_{i=1}^{n} y_i \left[\frac{K\left(\frac{x-x_i}{h_x}\right)}{\sum_{i'=1}^{n} K\left(\frac{x-x_{i'}}{h_x}\right)} \right]. \quad (51)$$

Choosing the band-width h_x of the kernel function is not an easy task. Usually it is done numerically by cross-validation. It is worth remarking that K-nearest neighbor smoothing is nothing but a kernel smoothing for which the kernel function is an unsymmetrical flat window spanning the range of the K-nearest neighbors of the point x. The kernel (51) is called Nadaraya-Watson kernel. Historically, [12] first introduced the window method density function estimation; then his work was pioneered by [13] and [14] in regression.

2.2. Additive Models

Recalling (23) and noticing that the function $f_i(X)$ is a scalar parametric function of the whole predictor shows that linear models are parametric additive models. By dropping the parametric assumption and letting each scalar function be a function of just one element of the predictor, i.e., X_i, allows defining a new nonparametric regression method, namely additive models, as:

$$h(x) = a + \sum_{i=1}^{p} f_i(X_i), \quad (52)$$

where the predictor is of dimension p. The response variable itself, Y, is modeled as in (20) by assuming zero mean and constant variance for the random component e. Then, $f_i(X_i)$ is fit by any smoothing method defined in Section 2.1. Every function $f_i(X_i)$ fits the value of the response minus the contribution of the other $p-1$ functions from the previous iteration. This is called the back-fitting algorithm described in [10]

2.3. Generalized Additive Models

Generalized additive models can be developed in a way analogous to how generalized linear models were developed above, i.e., by working with a transformation of the response variable, hence the name generalized additive models (GAM). Equation (52) describes the regression function as an additive model; alternatively it can be described through another link function:

$$g\left(h(x)\right) = a + \sum_{i=1}^{p} f_i(X_i) \quad (53)$$

Again, if a *logit* function is used the model can be used for classification exactly as was done in the case of generalized linear models. Rewriting the score equations (44) for the

GAM, using the posterior probabilities as the response variable, produces the nonparametric classification method using the GAM. Details of fitting the model can be found in [15].

2.4. Projection Pursuit Regression

Projection Pursuit Regression (PPR), introduced by [16], is a direct attack on the dimensionality problem, since it considers the regression function as a summation of functions, each of which is a function of a projection of the whole predictor onto a direction (specified by some unit vector). Formally it is expressed as:

$$h(x) = \sum_{i=1}^{k} g_i(a_i' x) \qquad (54)$$

The function g_i for every selection for the direction a_i is to be fit by a smoother in the new single variable $a_i' x$. It should be noted that (54) assumes that the function $g_i(a_i' X)$, named the *ridge function*, is constant along any direction perpendicular to a_i. Fitting the model is done by iteratively finding the best directions a_i's that minimize(s) the residual sum square of errors, hence the name pursuit. Details of fitting the model and finding the best projection directions can be found in [16] and [10].

In (54) by deliberately setting each unit vector a_i to have zero components except $a_{ii} = 1$, reduces the projection pursuit method to additive models. Moreover, and interestingly as well, by introducing the *logit* link function to the regression function $h(x)$ in (54) suits the classification problem exactly as done in the GAM. This turns out to be exactly the same as the single-hidden-layer neural network, as will be presented in the next section.

2.5. Neural Networks

Neural Networks (NN) have evolved in the engineering community since the 1950s. As illustrated in Figure 1, a neural network can be considered as a process for modeling the output in terms of a linear combination of the inputs.

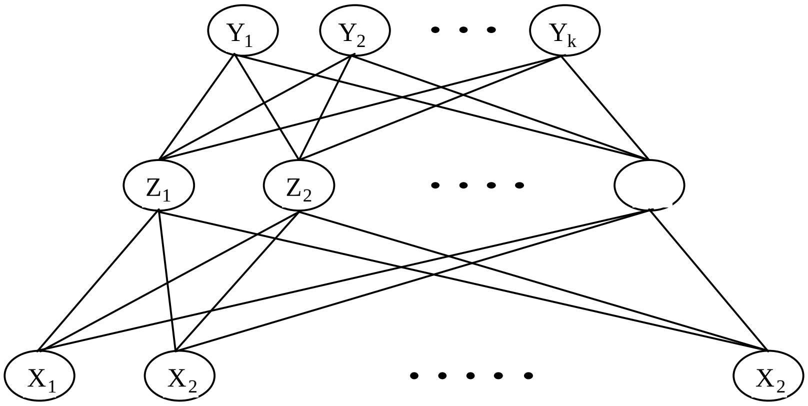

Figure 1 **Schematic diagram for a single hidden layer**

The set of p input features, i.e., the predictor components $X_1,\dots,X_p$, are weighted linearly to form a new set of M arguments, $Z_1,\dots,Z_M$, that go through the sigmoid function s. The output of the sigmoid functions accounts for a hidden layer consisting of M intermediate values. Then these M hidden values are weighted linearly to form a new set of K arguments that go through the final output functions whose output is the response variables $Y_1,\dots,Y_K$. This can be expressed mathematically in the form:

$$Z_m = s(a_{om} + a'_m X), \quad m = 1,2,\dots,M, \tag{55}$$

$$Y_k = f_k\left(b_{0k} + \sum_{m=1}^{M} b_{mk} Z_m\right), \quad k = 1,2,\dots,K \tag{56}$$

Figure 2 shows the function under different values of a (called learning rate below). The sigmoid function is defined by:

$$s(m) = \frac{1}{1 + e^{-m}} \tag{57}$$

Equation (56) shows that if the function f is chosen to be the identity function, i.e., $f(m) = m$, the neural network is simply a special case of the projection pursuit method defined in (54), where the sigmoid function has been explicitly imposed on the model rather than being developed by any smoothing mechanism as in PPR. This is what is done when the output of the network is quantitative. When it is categorical, i.e., the case of classification, the contemporary trend is to model the function f as:

$$f_k(m_k) = \frac{e^{m_k}}{\sum_{k'=1}^{K} e^{m_{k'}}} \tag{58}$$

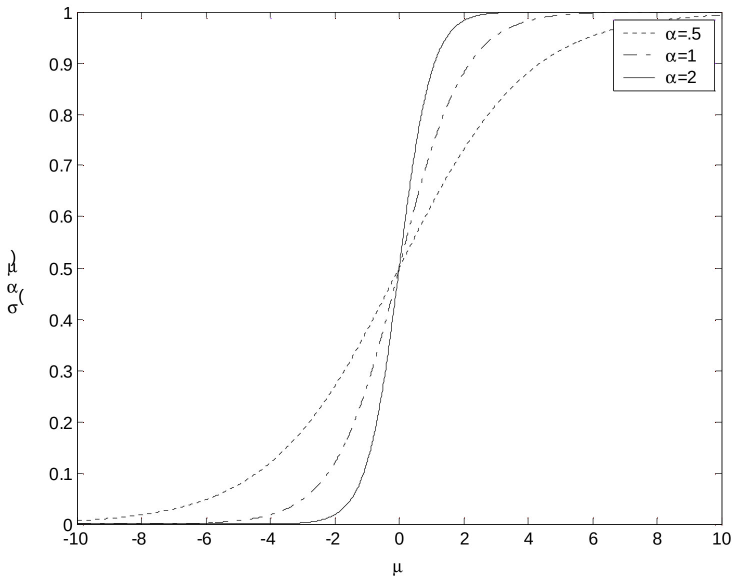

Figure 2 **Sigmoid function under different learning rate** a

In that case each output node models the posterior probability $\Pr[w_k \mid X]$, which is exactly what is done by the multi-logistic regression link function defined in (35). Again, the model will be an extension to the generalized additive models as defined at the end of Section 2.3. Excellent references for neural networks are [17] and [18]. We conclude this section by quoting the following statement from [10]:

"*There has been a great deal of hype surrounding neural networks, making them seem magical and mysterious. As we make clear in this section, they are just nonlinear statistical models, much like the projection pursuit regression model discussed above.*"

3. Computational Intelligence

The term computational intelligence was first coined by [19] and [20]:

"*A system is computationally intelligent when it: deals only with numerical (low-level) data, has a pattern recognition component, and does not use knowledge in the AI (Artificial Intelligence) sense; and additionally, when it (begins to) exhibit (i) computational adaptivity; (ii) computational fault tolerance; (iii) speed approaching human-like turnaround, and (iv) error rates that approximate human performance.*"

Since that time the term Computational Intelligence (CI) has been accepted as a generic term to the field that combines Neural Networks, Fuzzy Logic, and Evolutionary Algorithms; see [21] and [22]. As a still-developing field, CI may incorporate other methodologies as a coherent part. In [23], the area of Swarm Detection is considered as a peer paradigm to the other three mentioned above.

In the spirit of what has been discussed in the preceding sections, these methods assume nothing about the data distributions; they try to approach the solution by merely dealing with the data, i.e., numbers (c.f. the definition above). Hence, the CI methods, from a purely statistical point of view, are considered as nonparametric methods. Section 2 illustrated, mathematically, how Neural Networks, a basic building block in the CI field, is a special case of the projection pursuit, a nonparametric regression method.

4. No overall Winner among All Methods

This statement is important enough to be emphasized under a separate title, even though it has been touched upon throughout previous sections. If there is no prior information for the joint distribution between the response and the predictor, and if there is no prior information about the phenomenon to which that regression or classification will be applied, there is no overall winner among regression or classification techniques. If one classification method is found to outperform others in some application, this is likely to be limited to that very situation or that specific kind of problem; it may be beaten by other methods for other situations. In the engineering community, this concept is referred to as the *No-Free-Lunch* Theorem [24 Sec. 9.2]. This situation holds because each method makes different assumptions about the application or the process being modeled, and not all real-life applications are the same. If one or more of the assumptions are not satisfied in a given application, the performance will not be optimal in that setting.

5. Curse of Dimensionality and Dimensionality Reduction

In general, smoothing is difficult to implement in higher dimensions. This is because for a fixed number of observations available, the volume size needed to cover a particular percentage of the total number of observations increases by a power law, and thus exponentially, with dimensionality. This makes it prohibitive to include the same sufficient number of observations within a small neighborhood, or bandwidth, for a sample case to smooth the response. E.g., consider a unit hyper-cube in the p-dimensional subspace containing uniformly distributed observations; the percentage of the points located inside a hyper-cube with side length l is l^p. This means, if the suitable band-width for a certain smoother is l, the effective number of sample cases in the p-dimensional problem will go as the power $1/p$. This deteriorates the performance dramatically for p higher than 3. This is why the additive model, Section 2.2, and its variants are expressed as summation of functions of just one dimension. This single dimension may be just a component of the predictor or a linear combination.

A very crucial sub-field in statistical learning is dimensionality reduction; alternatively it is called feature selection in the engineering community. Qualitatively speaking, this means selecting those predictor components that best summarize the relationship between the response and predictor. In real-life problems, some features are statistically dependent on others; this is referred to as multi-collinearity. On the other hand, there may also be some components that are statistically independent with the response. These add no additional information to the problem at all; thus they serve only as a source of noise. This is a rapidly maturing sub-field. A remarkable publication in the statistics literature in this regard is that by [25]. It introduces the Sliced Inverse Regression (SIR), in which each predictor component is regressed on the response; hence the name inverse regression. In that sense, the problem is reduced from regressing a single response on a p-dimensional predictor to regressing p-responses on a single-dimensional new predictor, which is far simpler than the former.

6. Unsupervised Learning

It should be noticed that the formal definition of the learning process, discussed thus far assumed the existence of a training data set, name it, $\mathbf{t} : \{t_i = (x_i, y_i)\}$. Each element t_i, or sample case, in this set has an already known value for the response variable; this is what enables the learning process to develop the relationship between the predictor and the response. This is what is called supervised learning. On the contrary, in some applications the available data set is described by $\mathbf{t} : \{t_i = x_i\}$ without any additional information. This situation is called unsupervised learning. The objective in such a situation is to understand the structure of the data from the available empirical probability distribution of the points x_i. For the special case where the data come from different classes, the data will be represented in the hyper p-dimensional subspace , to some extent, as disjoint clouds of data. The task in this case is called clustering, i.e., trying to identify those classes that best describe, in some sense, the current available data. More formally, if the available data set is $\mathbf{X}$, the objective is to find the class vector $W = [w_1, \ldots, w_k]'$ such that a criterion $J(\mathbf{X}, W)$ is minimized:

$$W = \operatorname{argmin} J(\mathbf{X}, W) \qquad (59)$$

Different criteria give rise to different clustering algorithms. More discussion on unsupervised learning and clustering can be found in [10, 24, 26].

7. Performance of Classification Rules

From what has been discussed until now, there is not any conceptual difference between regression and classification for the problem of supervised learning. Abstractly, both aim to achieve the minimum risk under a certain loss function for predicting a response from a particular predictor. If the special case of classification is considered, there should be some measure to assess the performance of the classification rule. Said differently, if several classifiers are competing in the same problem, which is better? One natural answer is to consider the risk of each classifier, as was defined in (10).

A special case of classification, which is of great interest in many applications, is binary classification, where the number of classes is just two. In that case the risk of each classifier is reduced to (19), which can be rewritten as:

$$R_{min} = c_{12}P_1e_1 + c_{21}P_2e_2 \tag{60}$$

where e_1 is the probability of classifying a case as belonging to class 2 when it belongs to class 1, and e_2 is vice versa.

In the feature subspace, the regions of classification have the dimensionality p, and it is very difficult to calculate the error components from multi-dimensional integration. It is easier to look at (17) as:

$$h(x) \overset{\omega_1}{\underset{\omega_2}{\gtrless}} th, \qquad \text{where} \tag{61}$$

$$h(x) = \log\frac{f_X(X = x \mid \omega_1)}{f_X(X = x \mid \omega_2)}, \tag{62}$$

$$th = \log\frac{\Pr\left[\omega_1\right](c_{22} - c_{21})}{\Pr\left[\omega_2\right](c_{11} - c_{12})}, \tag{63}$$

and $h(X)$ is called the log-likelihood ratio. Now the log-likelihood ratio itself is a random variable whose variability comes from the feature vector X, and has a PDF conditional on the true class. This is shown in Figure 3. It can be easily shown that the two curves in Figure 3 cross at $h(X) = 0$, where the threshold is zero. In this case the two error components, appearing in (60), are written equivalently as:

$$e_1 = \int_{-\infty}^{th} f_h\,(h(x) \mid \omega_1)dh(x), \tag{64}$$

$$e_2 = \int_{th}^{\infty} f_h\,(h(x) \mid \omega_2)dh(x) \tag{65}$$

Now assume the classifier is trained under the condition of equal prevalence and costs, i.e., the threshold is zero. In other environments there will be different a priori probabilities yielding to different threshold values. The error is not a sufficient metric now, since it is function of a single fixed threshold. A more general way to assess a classifier is provided by the Receiver Operating Characteristic (ROC) curve. This is a plot for the two components of error, e_1 and e_2 under different threshold values. It is conventional in medical imaging to refer to e_1 as the False Negative Fraction (FNF), and e_2 as the False Positive Fraction (FPF). This is because diseased patients typically have a higher output value for a test than non-diseased patients. For example, a patient belonging to class 1 whose test output value is less than the threshold setting for the test will be called “test negative” while the patient is in fact in the diseased class. This is a false negative decision; hence the name FNF. The situation is reversed for the other error component.

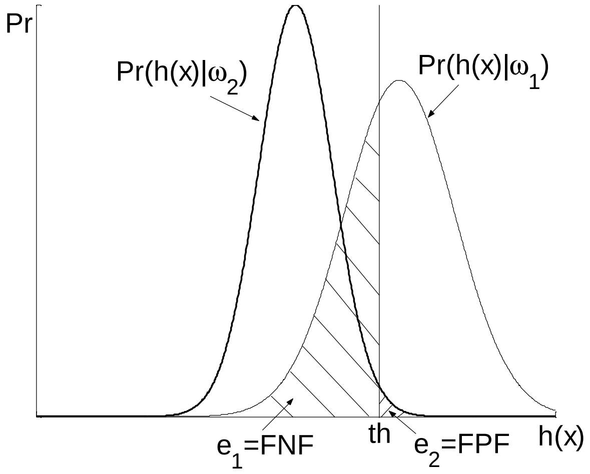

Figure 3. **The probability of loglikelihood ratio conditional under each class. The two components of error are indicated as the FPF and FNF, the conventional terminology in medical imaging.**

Since the classification problem now can be seen in terms of the log-likelihood, it is apparent that the error components are integrals over a particular PDF. Therefore the resulting ROC is a monotonically non-decreasing function. A convention in medical imaging is to plot the $TPF\ (= 1 - FNF)$ vs. the FPF. In that case, the farther apart the two distributions of the log-likelihood function from each other, the higher the ROC curve and the larger the area under the curve (AUC). Figure 4 shows ROC curves for two different classifiers.

The first one performs better since it has a lower value of e_2 at each value of e_1. Thus, the first classifier unambiguously separates the two classes better than the second one. Also, the AUC for the first classifier is larger than that for the second one. AUC can be thought of as one summary metric for the ROC curve.

Formally the AUC is given by:

$$AUC = \int_0^1 TPF\ d(FPF). \tag{66}$$

If two ROC curves cross, this means each is better than the other for a certain range of the threshold setting, but it is worse in another range. In that case some other performance measure can be used, such as the partial area under the ROC curve in a specified region.

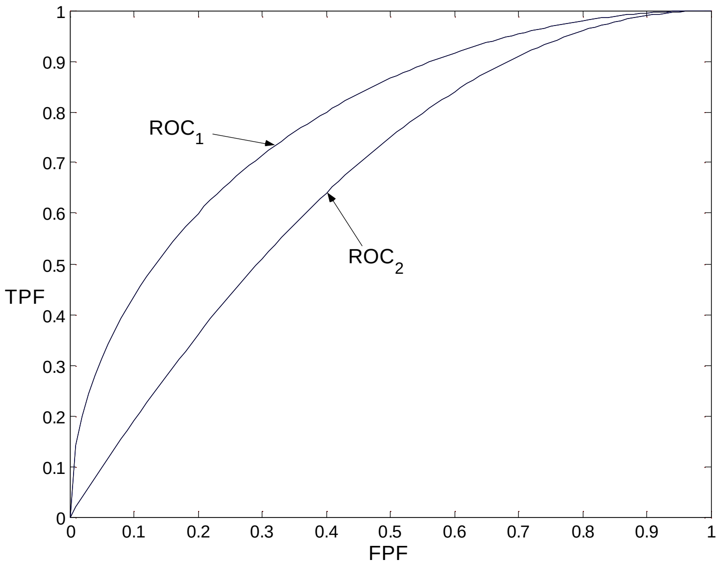

Figure 4. ROC curves for two different classifiers. ROC$_1$ is better than ROC$_2$, since for any error component value, the other component of classifier 1 is less than that one of classifier 2.

The two components of error in (60), or the summary measure AUC in (66), are the parametric forms. That is, these measures can be calculated by these equations if the posterior probabilities are known parametrically, e.g., in the case of the Bayes classifier or by parametric regression techniques as in Section 1.

On the contrary, if the posterior probabilities are not known in a parametric form, the error rates can be estimated only numerically from a given data set, called the testing data set. This is done by assigning equal probability mass for each sample case, since this is the Maximum Likelihood Estimation (MLE) for the probability mass function under the nonparametric distribution. This can be proven by maximizing the likelihood function:

$$L(F) = \prod_{i=1}^{n} p_i \tag{67}$$

under the constraint $\Sigma_i p_i = 1$. The likelihood (67) can be rewritten, by considering this constraint, using a Lagrange multiplier as:

$$L(F) = \prod_{i=1}^{n} p_i + \lambda \left(\sum_{i=1}^{n} p_i - 1 \right) \tag{68}$$

The likelihood (68) is maximized by taking the first derivative and setting it to zero to obtain:

$$\frac{\partial L(F)}{\partial p_j} = \prod_{i \neq j} p_i + \lambda \overset{\text{set}}{=} 0, \quad j = 1, \ldots, n \tag{69}$$

These n equations along with the constraint $\sum_i p_i = 1$ can be solved straightforwardly to give:

$$\hat{p}_i = \frac{1}{n}, \quad i = 1,\ldots,n \tag{70}$$

That is, the nonparametric MLE of the distribution will be

$$\hat{F} : \text{mass } \frac{1}{n} \text{ on } t_i,\ i = 1,\ldots,n, \tag{71}$$

where n is the size of the testing data set. In this case (2) will be reduced to:

$$\hat{R}(h) = E_{\hat{F}}[L(Y, h(X))] \tag{72}$$

$$= \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, h(x_i)\right) \tag{73}$$

where the expectation has been taken over the empirical distribution $\hat{F}$ of the variable. In the case of classification, (72) can be reduced further to:

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} c_{i,h(x_i)} \tag{74}$$

In the special case of zero loss for correct decisions in binary classification, (74) reduces further to

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(c_{12} I_{\hat{h}(x_i|w_1)<th} + c_{21} I_{\hat{h}(x_i|w_2)>th}\right) \tag{75}$$

$$= \frac{1}{n}\left(c_{21}\hat{e}_1 n_1 + c_{21}\hat{e}_2 n_2\right) \tag{76}$$

$$= c_{21}\widehat{FNF}\,\hat{P}_1 + c_{21}\widehat{FPF}\,\hat{P}_2 \tag{77}$$

which is the nonparametric approximation to (60) and (64). The indicator function I is defined in (50). The values n_1 and n_2 are the sizes of class-1 sample and class-2 sample respectively, and $\hat{P}_1$ and $\hat{P}_2$ are the estimated a priori probabilities. The function $\hat{h}(x_i)$ is the estimated log-likelihood ratio at case t_i obtained from estimating the posterior probabilities with any of the nonparametric classification methods (Section 2). In the case of $c_{12} = c_{21} = 1$, the so-called "0-1 loss function", the risk is called simply the error rate or (Probability of Misclassification (PMC)).

The two components, $1 - \widehat{FNF}$ and $\widehat{FPF}$ give one point on the empirical (estimated) ROC curve. To draw the complete curve in the nonparametric situation, the estimated log-likelihood is calculated for each point of the available data set. Then all possible thresholds are considered in turn, i.e., the threshold values between every two successive estimated log-likelihood values. At each threshold value a point on the ROC curve is calculated. Then the AUC can be calculated numerically from the empirical ROC curve using the trapezoidal rule:

$$\hat{AUC} = \frac{1}{2}\sum_{i=2}^{n_{th}} (FNF_i - FNF_{i-1})(TPF_i + TPF_{i-1}) \qquad (78)$$

where n_{th} is the number of threshold values taken over the data set. By plotting the empirical ROC curve, it is easy to see that the AUC obtained from the trapezoidal method is the same as the Mann-Whitney statistic—which is another form of the Wilcoxon rank-sum test [H27]—defined by:

$$\hat{AUC} = \frac{1}{n_1 n_2}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\psi\left(\hat{h}(x_i \mid w_1), \hat{h}(x_j \mid w_2)\right), \qquad (79)$$

$$\psi(a,b) = \begin{cases} 1 & a < b \\ 0 & a < b \end{cases} \qquad (82)$$

The equivalence of the area under the empirical ROC and the Mann-Whitney-Wilcoxon statistic is the basis of its use in the assessment of diagnostic tests; see [28]. [29] has recommended it as a natural summary measure of detection accuracy on the basis of signal-detection theory. Applications of this measure are widespread in the literature on human and computer-aided diagnosis in medical imaging, e.g., [30]. In the field of machine learning, [31] has recommended it as the preferred summary measure of accuracy when a single number is desired. These references also provide general background and access to the large literature on the subject.

It has been mentioned above that in the nonparametric situation these performance measures are estimated from a single given data set, i.e., the testing data set or, less formally, the testers. But as long as the distribution is unknown it is not only impossible to calculate these measures parametrically, but it is also impossible to generate, by simulation, testing data sets on which these metrics can be estimated. In that case the classifier might be trained and its performance measure estimated from the same training data set. This estimation will be a random variable whose randomness comes from the finite training data set $\mathbf{t}$. That is, under different data sets even of the same size, the estimate will vary. Therefore it is not sufficient to assess a classifier performance by estimating its mean, either error or AUC, without estimating the variability.

In general, the fundamental population parameters of interest are the following: The true performance $AUC_{\mathbf{tr}}$ conditional on a particular training data set $\mathbf{tr}$ of a specified size but over the population of testing data sets—as if we trained on $\mathbf{tr}$ then tested on infinite number of observations; the expectation of this performance over the population of training data sets of the same size, $E_{\mathbf{tr}}AUC_{\mathbf{tr}}$; and the measure of variability of this performance over the population of training data sets, of the same size, $Var_{\mathbf{tr}}AUC_{\mathbf{tr}}$. Estimators of these parameters, respectively, $\hat{AUC}_{\mathbf{tr}}$, $\hat{E}_{\mathbf{tr}}AUC_{\mathbf{tr}}$, and $\hat{Var}_{\mathbf{tr}}AUC_{\mathbf{tr}}$, can be obtained in several ways. Parametric estimates can be obtained by modeling the underlying distributions of the samples, e.g., as in [26].

If the distributions of the samples are either unknown or not readily modeled, then this is a problem of nonparametric estimation. There are several traditional approaches to using the available data in this estimation task. One approach is to have a common data set that is used for training and testing; this approach often includes various resampling strategies, including cross-validation and bootstrapping [32-34]. Another approach is to maintain what might be called the traditional data hygiene of two independent data sets, the training data set $\mathbf{tr}$ (simply called trainers), and the testing data set $\mathbf{ts} = \{t_i : t_i = (x_i, y_i), i = 1, \ldots, n_{\mathbf{ts}}\}$ (simply called testers). Therefore, the reader should keep in mind the fact that the three estimators above are functions of both $\mathbf{tr}$ and $\mathbf{ts}$ although they are not $\mathbf{ts}$-subscripted.

The first two of these estimators, $\widehat{\mathrm{AUC}}_{\mathbf{tr}}$ and $\widehat{\mathrm{E}}_{\mathbf{tr}}\mathrm{AUC}_{\mathbf{tr}}$, were discussed, along with their variances, in [35] and [36], where there was only one available data set for training and testing. In that paradigm, training was pursued on different bootstrap replications from the available data set while testing was done by testing on the remaining observations that did not appear in the bootstrap replications. This technique was developed in [33] and [37], and their performance index was the total error, rather than the AUC.

There are some situations, e.g., in several public-policy-making or regulatory settings, in which it could be highly recommended, or even mandatory, that the training and testing sets be isolated as in the so-called traditional hygiene. This technique is analyzed in [38].

It is worth mentioning that assessment in terms of the AUC as the index (or measure) is straightforward to be extended to other summary measures of performance such as the partial area under the curve (PAUC) in some specified region of interest; see [39].

8. Conclusion and Advice for Practitioners

In this article, the importance of statistical learning is stressed through demonstrating examples from different areas and applications. The mathematical foundations of the field, along its different methods of design, have been motivated. Last section was dedicated to the assessment problem of a designed classifier. Bearing in mind that this article is intended to be a tutorial article on the field, important and fundamental references have been cited, wherever necessary, for readers interested in more elaboration.

Many practitioners in the field leverage some methods, in designing their classifier, without having enough insight; this leads to fallacies in results or conclusions. Example of this is the exaggerated use of neural networks with multiple layers leading to overtraining. Another pitfall is using a small size training data set with respect to the dimensionality of the problem. This is always the case in some fields, e.g., DNA microarrays. However, a more elaborate assessment phase should follow the design phase in these ill-posed applications. A third pitfall is assessing classifiers in only the mean performance ignoring the variance arousing from the finite sample size. Overlooking these conceptual and mathematical foundations—which is always observed in the field—in both design and assessment, drives practitioners to, at best, flukes; while their findings and conclusions, sometimes, are fragile.

References:

1. Casella, G. and R.L. Berger, *Statistical inference*. 2nd ed. Duxbury advanced series. 2002, Australia; Pacific Grove, CA: Duxbury/Thomson Learning. xxviii, 660 p.

2. Billingsley, P., *Probability and measure*. 3rd ed. Wiley series in probability and mathematical statistics. 1995, New York: Wiley. xii, 593 p.

3. Graybill, F.A., *Theory and application of the linear model*. 1976, North Scituate, Mass.: Duxbury Press. xiv, 704 p.

4. Lehmann, E.L. and G. Casella, *Theory of point estimation*. 2nd ed. Springer texts in statistics. 1998, New York: Springer. xxvi, 589 p.

5. Bowerman, B.L. and R.T. O'Connell, *Linear statistical models: an applied approach*. 2nd ed. Duxbury advanced series in statistics and decision sciences. 1990, Boston: PWS-Kent Pub. Co. xvi, 1024 p.

6. Christensen, R., *Plane answers to complex questions: the theory of linear models*. 3rd ed. Springer texts in statistics. 2002, New York: Springer. xix, 473 p.

7. Rencher, A.C., *Linear models in statistics*. Wiley series in probability and statistics. 2000, New York: Wiley. xviii, 578 p.

8. Lehmann, E.L. and J.P. Romano, *Testing statistical hypotheses*. 3rd ed. Springer texts in statistics. 2005, New York: Springer. xiv, 784 p.

9. Anderson, T.W., *An introduction to multivariate statistical analysis*. 3rd ed. Wiley series in probability and statistics. 2003, Hoboken, N.J.: Wiley-Interscience. xx, 721 p.

10. Hastie, T., R. Tibshirani, and J.H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. 2001, New York: Springer. xvi, 533 p.

11. Efron, B., *The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis.* Journal of the American Statistical Association, 1975. **70**(352): p. 892-898.

12. Parzen, E., *On Estimation of a Probability Density Function and Mode.* The Annals of Mathematical Statistics, 1962. **33**(3): p. 1065-1076.

13. Nadaraya, E.A., *On Estimating Regression.* Theory of Probability and Its Applications, 1964. **9**(1): p. 141-142.

14. Watson, E.S., *Smooth Regression Analysis.* Sankhy\={a}: The Indian Journal of Statistics, 1964. **Series A, 26**: p. 359-372.

15. Hastie, T. and R. Tibshirani, *Generalized additive models*. 1st ed. Monographs on statistics and applied probability; 43. 1990, London; New York: Chapman and Hall. xv, 335 p.

16. Friedman, J.H. and W. Stuetzle, *Projection Pursuit Regression.* Journal of the American Statistical Association, 1981. **76**(376): p. 817-823.

17. Bishop, C.M., *Neural networks for pattern recognition*. 1995, Oxford; New York: Clarendon Press; Oxford University Press.

18. Ripley, B.D., *Pattern recognition and neural networks*. 1996, Cambridge; New York: Cambridge University Press. xi, 403 p.

19. Bezdek, J.C., *On the relationship between neural networks, pattern recognition and Intelligence.* The International Journal of Approximate Reasoning, 1992. **6**: p. 85-107.

20. Bezdek, J.C., *What is computational intelligence?* in *Computational intelligence:*

imitating life, J.M. Zurada, R.J. Marks, and C.J. Robinson, Editors. 1994: New York. p. 1-12.

21. Schwefel, H.-P., I. Wegener, and K. Weinert, eds. *Advances in computational intelligence: theory and practice*. Natural computing series. 2003, Springer: Berlin; New York. ix, 323 p.

22. Zimmermann, H.-J.u.r., et al., eds. *Advances in Computational Intelligence and Learning: Methods and Applications*. International series in intelligent technologies; 18. 2002, Kluwer Academic Publishers: Boston. xvi, 511 p.

23. Engelbrecht, A.P., *Computational intelligence: an introduction*. 2002, Chichester, England; Hoboken, N.J.: J. Wiley \& Sons. xvi, 288 p.

24. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern classification*. 2nd ed. 2001, New York: Wiley. xx, 654 p.

25. Li, K.-C., *Sliced Inverse Regression for Dimension Reduction.* Journal of the American Statistical Association, 1991. **86**(414): p. 316-327.

26. Fukunaga, K., *Introduction to statistical pattern recognition*. 2nd ed. Computer science and scientific computing. 1990, Boston: Academic Press. xiii, 591 p.

27. H\'{a}jek, J., z.v.e.k. \v{S}id\'{a}k, and P.K. Sen, *Theory of rank tests*. 2nd ed. Probability and mathematical statistics. 1999, San Diego, Calif.: Academic Press. xiv, 435 p.

28. Hanley, J.A. and B.J. McNeil, *The meaning and use of the area under a receiver operating characteristic ({ROC}) curve.* Radiology, 1982. **143**(1): p. 29-36.

29. Swets, J.A., *Indices of Discrimination or Diagnostic Accuracy: Their {ROC}s and Implied Models.* Psychological Bulletin, 1986. **99**: p. 100-117.

30. Jiang, Y., et al., *Improving breast cancer diagnosis with computer-aided diagnosis.* Academic Radiology, 1999. **6**(1): p. 22-33.

31. Bradley, A.P., *The use of the area under the {ROC} curve in the evaluation of machine learning algorithms.* Pattern Recognition, 1997. **30**(7): p. 1145.

32. Efron, B., *Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation.* Journal of the American Statistical Association, 1983. **78**(382): p. 316-331.

33. Efron, B. and R. Tibshirani, *Improvements on Cross-Validation: The $.632+$ Bootstrap Method.* Journal of the American Statistical Association, 1997. **92**(438): p. 548-560.

34. Stone, M., *An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion.* Journal of the Royal Statistical Society. Series B (Methodological), 1977. **39**(1): p. 44-47.

35. Yousef, W.A., R.F. Wagner, and M.H. Loew. *Comparison of Non-Parametric Methods for Assessing Classifier Performance in Terms of {ROC} Parameters*. in *Applied Imagery Pattern Recognition Workshop, 2004. Proceedings. 33rd; IEEE Computer Society*. 2004.

36. Yousef, W.A., R.F. Wagner, and M.H. Loew, *Estimating the Uncertainty in the Estimated Mean Area Under the {ROC} Curve of a Classifier.* Pattern Recognition Letters, 2005. **26**(16): p. 2600-2610.

37. Efron, B. and R. Tibshirani, *Cross Validation and the Bootstrap: Estimating the Error Rate of a Prediction Rule.* Technical Report 176, Stanford University, Department of Statistics, 1995.

38. Yousef, W.A., R.F. Wagner, and M.H. Loew, *Assessing Classifiers from Two Independent Data Sets Using {ROC} Analysis: A Nonparametric Approach.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006. **28**(11): p. 1809-1817.

39. Yousef, W.A., R.F. Wagner, and M.H. Loew, *The Partial Area under the ROC Curve: Its Properties and Nonparametric Estimation for Assessing Classifier Performance.* A Manuscript Submitted to Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005.

40. Yousef, W.A., *Assessment of statistical classification rules: implications for computational intelligence*, in *Electrical and Computer Engineering*. 2006, The George Washington University: Washington DC.