Supervised Classification, a Probabilistic Approach

Pierre Comon

Thomson-Sintra,
BP 157, F-06903 Sophia-Antipolis Cedex *

Abstract. In this tutorial, some classical tools of data analysis –and others less well known– are surveyed in detail, that can be applied to supervised classification, and in particular to learning with neural networks.
Keywords. Bayes, Performance, Kernel, Parzen, RBF, Density, Asymptotic theorems, Cross-validation, Clustering, Small samples.

1. Introduction

Supervised learning is performed usually by minimizing some objective function, that can be an error e in fitting an arbitrary desired function, or a misclassification rate ϵ. The computation of the latter often raises difficulties. In fact, if the Bayesian approach is assumed, densities need to be estimated, and this can become quite difficult in large dimensions. There are direct ways of computing the total misclassification rates, without going through the Bayesian formalism, but they are computationally heavy. Yet, they can still be appropriate if the best achievable performance are sought under memory constraints (*e.g.* low-cost neural networks).

In this paper, links between e-minimization and ϵ-minimization are pointed out under different aspects. Emphasis is given on kernel estimator of densities, which seem to perform very well in practical experiments. The principles exposed give birth essentially to off-line algorithms, and recursive algorithms are not presented for reasons of space.

A side goal of this paper is to give a flavour of the theory on which the developments of the *Elena project* were based, and to report some choices that have been made. The implementation of the subsequently presented tools and algorithms in the packlib software is being currently completed .

Part of this work has been funded by the ESPRIT-BRA project 6891, supported by the Commision of the European Communities.
*Also with I3S–CNRS, 250 av A.Einstein, Sophia-Antipolis, F-06560 Valbonne. comon@mimosa.unice.fr

Organization of the paper

The first section aims mainly at defining notation, and introduces the confusion matrix. Section 3. summarizes how performance has been chosen to be evaluated. Section 4. reports a result that justifies the use of an ϵ-criterion rather than a fitting e-criterion. Fixed and variable kernel estimators are described in section 5., and a suboptimal implementation with clusters is described in section 6. After some computational considerations in section 7., we close with some comments on a current work in progress in section 8.

2. Probabilistic framework

The classification problem consists of building a mapping ϕ from a set of patterns (observations), $\mathcal{E}$, to a set of classes, $\mathcal{F}$. It is assumed throughout this paper that patterns are real valued and of dimension d. In other words, $\mathcal{E} = \mathbb{R}^d$. Thus, any pattern x in $\mathcal{E}$ is wished to be associated with a class $\omega_{j(x)} \in \mathcal{F}$ by this mapping.

In the context of supervised classification, a set of examples $\mathcal{A}(N) = \{(x(n), \omega_{j(n)}), 1 \le n \le N\}$ is given, so that mapping ϕ is apparently known at a finite number of points. This set of input-output pairs is the *learning set*. In the Bayesian approach, the mapping built may not assign all the patterns from the learning set to their true class. This freedom allows for instance to handle learning sets where some equal (or very close) patterns appear with different class labels.

If patterns $\{x(n), 1 \le n \le N\}$ in the learning set are known to belong to classes $\omega_j, 1 \le j \le K$, it is natural to choose as output space $\mathcal{F}_A = \{\omega_j, 1 \le j \le K\}$. However, as will be recalled shortly, there are reasons to add two other classes: one for ambiguities, and one for rejections. As a consequence, mapping ϕ is actually defined from $\mathcal{E}$ onto $\mathcal{F} = \mathcal{F}_A \cup \{\omega_{K+1}, \omega_{K+2}\}$.

In a probabilistic framework, it is assumed that all patterns belonging to the same class are *independently drawn* from the same underlying distribution. In this paper, it is assumed that this distribution admits a density, denoted $p(x|\omega_j)$.

Assuming uniform losses, the Bayesian approach allows to build the mapping that minimizes the total number of misclassifications, provided the conditional densities $p(x|\omega_j)$ and the priors $P_j = P(\omega_j)$ are known. More precisely, since the output space is discrete, the mapping ϕ defines K disjoint domains $D_j = \phi^{-1}(\omega_j)$ in the input space $\mathcal{E}$, so that in each D_j any pattern is assigned class ω_j. Then a well-known way of writing the Bayesian risk function is (still assuming uniform losses):

$$R = \sum_{\substack{i,j=1 \\ i \neq j}}^{K} C_{ij}(\phi), \quad C_{ij}(\phi) \stackrel{\text{def}}{=} P_i \int_{u \in D_j} p(u|\omega_i)\, du. \tag{1}$$

The integral corresponds to the probability that ϕ assigns the class ω_j to a pattern x whereas its true class is ω_i. The probabilities of error $C_{ij}(\phi)$ can be arranged in a $K \times K$ matrix, often called the *confusion matrix*.

Then it has been shown [12] [14] [17] [2] that minimizing R has the following solution: the class $\omega_{j(x)}$ is assigned to observation $x \in \mathcal{E}$ if and only if:

$$j(x) = Arg \underset{1 \leq i \leq K}{\mathrm{Max}} \{P_i \, p(x|\omega_i)\} = Arg \underset{1 \leq i \leq K}{\mathrm{Max}} \{p(x, \omega_i)\}. \tag{2}$$

The decision is unambiguous at a point x of $\mathcal{E}$ if all $p(x, \omega_i)$'s are different. But if the largest value is reached by several classes, there is an ambiguity. One can distinguish between two kinds of ambiguity:

- All values of $p(x, \omega_i)$ are very small, in which case one cannot reasonably assign to x one of the classes present in the learning set. In that case, x is assigned the *reject* class, ω_{K+1}. This occurs on a domain denoted D_{K+1} in the input space $\mathcal{E}$.
- Otherwise, there are at least two large and equal values of $p(x, \omega_i)$ for some i's. Then, there is an ambiguity of decision between those classes, and class ω_{K+2} is (provisionally) assigned to pattern x. This occurs in a domain D_{K+2} of $\mathcal{E}$, that is the union of all other domains boundaries.

It is sometimes convenient to add two columns to the confusion matrix, one for ambiguities, and the other for rejections. Then we end up with a $K \times K + 2$ matrix whose rows sum up to one, because now $\cup_{j=1}^{K+2} D_j = \mathcal{E}$, and $\int_{\mathcal{E}} p(u|\omega_i) \, du = 1, \forall i$.

Now, it is clear that knowing a mapping on a finite set will never provide the complete definition of the mapping on $\mathcal{E}$ without further information. That's why supervised classification is usually carried out by assuming –sometimes implicitly– a parametric model, either on the classifying rules (as in neural networks), or on the conditional densities (as in Bayesian approaches). When the number of parameters is very large, the model is (somewhat abusively) referred to as *non parametric*. We shall see in section 5. one of these non parametric models.

3. Classification errors

3.1. Apparent confusion matrix

By definition, the best confusion matrix is attained by the exact Bayesian classifier. But in practice, domains D_j are only estimated by domains $\hat{D}_j$, and we are talking about performances of an estimated classifier $\hat{\phi}$. The confusion matrix corresponding to *exact* performances of the *estimated* classifier is given by:

$$C_{ij}(\hat{\phi}) = \int_{\hat{D}_j} p(u|\omega_i) \, du. \tag{3}$$

Similarly, exact performances cannot in general be calculated, because densities $p(u|\omega_i)$ must be replaced by estimates, that are denoted here $\check{p}(u|\omega_i)$. Here estimates are denoted differently on purpose, because there is no obligation to use the same density estimates to determine the classifier, and to compute its performances. Thus the estimated confusion takes the expression:

$$\check{C}_{ij}(\hat{\phi}) = \int_{\hat{D}_j} \check{p}(u|\omega_i)\, du, \tag{4}$$

Since estimates $\check{p}(u|\omega_i)$ are intended to be integrated on an arbitrary domain, it is necessary to choose them in such a way that this computation is easy to carry out, bearing in mind that only the integrated value is important (a local accuracy is superflous). In fact, for computational tractability, it is quasi always assumed that the density $\check{p}(u|\omega_i)$ has the simple form:

$$\check{p}(u|\omega_i) = \alpha_i \sum_{x(m)\in\omega_i} \delta(u - x(m)), \tag{5}$$

where $\delta(u)$ denotes the Dirac distribution, and α_i is a coefficient chosen so that the estimated density sum up to one. It may be checked that this calculation of the confusion reduces to a mere counting of the misclassified patterns.

If estimates $\hat{\phi}$ and $\check{p}(u|\omega_i)$ are using the same data, then the resulting confusion matrix is called *apparent*, because it is too optimistic. Some authors refer to this computation as the *Resubstitution method* [15]. It is well known that these two estimates should be independent for the confusion to be unbiased. In particular, this is achieved if the classifier and the performances are computed by using two disjoint sets of data. It is talked about cross-validation procedures.

3.2. Computation by cross-validation

Even if this family of procedures is quite well known, it may be useful to say a word on this topic. The simplest procedure is the Holdout. The available data are partitioned into two sets, one dedicated to learning, and the other to performance evaluation. The drawback is that part of the data is not used at all for learning. To face this objection, one can run several Holdouts, and average the performances obtained, but this becomes very costly.

There is however a case where the Averaged Holdout is not that costly, namely when the performance set is reduced to a single pattern. In that case, the learning is made on $N-1$ patterns, so that one can hope to have almost the best possible classifier. Since the partition is now $\{N-1\}\{1\}$, there are only N possible distinct runs to perform and to average. If the N partitions are tested, the procedure is referred to as the *totally Averaged Leave One Out* (ALOO). This seems to be the best way to fully use the information contained in the data, without biasing the performance estimation. Moreover, as pointed out by Fukunaga, there is often the possibility to derive the ALOO performances from those obtained by the Resubstitution [14] with little extra work.

There are close links between the ALOO and the general theory of the Jackknife. Efron has also pointed out that the Bootstrap may also be viewed as an extension of the Jackknife [15].

3.3. Confidence intervals

Denote $\mathcal{A}_i(N)$ the subset of the learning set $\mathcal{A}(N)$ that contains patterns belonging to class ω_i, and N_i its cardinality, $\sum N_i = N$. In addition, denote M_{ij} the number of patterns from $\mathcal{A}_i(N)$ that have been misclassified in class ω_j. The entry C_{ij} of the confusion matrix can be estimated by the ratio:

$$\check{C}_{ij} = \frac{M_{ij}}{N_i}. \tag{6}$$

In order to acces to confidence intervals for matrix $\check{C}$, one can remark that each M_{ij} follows a binomial distribution:

$$prob(M_{ij} = m) = \begin{pmatrix} N_i \\ m \end{pmatrix} C_{ij}^m \, (1 - C_{ij})^{N_i - m}. \tag{7}$$

There exists sophisticated ways of approximating the quantiles of $\check{C}_{ij}$, based on this distribution [20] and standard tables can be used. Note that, if N_i is large, one can reasonably assimilate $\check{C}_{ij}$ to a Gaussian variable with mean C_{ij} and variance C_{ij}^2, even if this approximation is very crude. In fact, an error of 5% or even 10% on the confidence interval is of little importance for our use. Nevertheless, if C_{ij} is close to 0 or 1, this approximation becomes too pessimistic. Strictly speaking, (7) is valid for a single holdout. Other more complete approaches include significance testing, but are not addressed in this paper.

As an example, if $N_i = 50$, and $\check{C}_{ij} = 0.2$, then true value of the confusion entry satisfies approximately $0.1 < C_{ij} < 0.3$ with a probability of 0.95, whereas for $N_i = 1000$, it satisfies $0.18 < C_{ij} < 0.22$.

4. Fitting errors

Assume each class ω_i is represented in $\mathcal{F}$ by an element z_i of an Hilbert space, so that we can admit that $\mathcal{F} \subset \mathbb{R}^m$ for some m. Assume also that the mapping $\phi(\cdot)$ is coded by a set of parameters W so that $\phi(\cdot) = \Phi(\cdot, W)$, where Φ is fixed. One approach of the problem is to search for a W so that $\Phi(\cdot, W)$ fits the input-output relations given by the learning set $\mathcal{A}(N)$ the best way, in the sense of the norm on $\mathcal{F}$:

$$W = Arg \operatorname*{Min}_{W} \frac{1}{N} \sum_{n=1}^{N} ||z_{i(n)} - \Phi(x(n), W)||^2. \tag{8}$$

We shall refer to this criterion as the output Minimum Quadratic Error (MQE). For instance, learning algorithms dedicated to the MultiLayer Perceptron answer that problem.

The first key remark to make is that the coding of outputs has a strong influence on the result obtained, and that it is completely arbitrary. Take an example if the number of classes is $K = 3$. Here are some possible choices:

$$\mathbf{i}: \{1, 2, 3\}, \quad \mathbf{ii}: \{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\}, \quad \mathbf{iii}: \{\begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}\} \tag{9}$$

and many others can be thought of. A natural question is to know whether the minimization of (8) would lead to the Bayesian solution, and with what output coding.

A partial answer has been given in [26], but a more complete one —but less well known— has been devised in [6]. The basic result is summarized by the following theorem proved in [26], and extended in [6] to general losses:

Theorem 1 *Denote N_i the number of patterns belonging to class ω_i in the learning set $\mathcal{A}(N)$. Assume that the absolute minimum in (8) is reached. Then $\Phi(\cdot, W)$ tends to the best approximation of the Bayesian solution as every N_i tends to infinity provided coding (9)***ii** *is chosen.*

This theorem shows that the power of the output error criterion (8) is rather limited, and also tends to say that the other output codings are inappropriate. In fact, other codings just yield other Bayesian solutions with different loss matrices, as proved in [6].

When the learning set is limited, it may be rather uninteresting to have such asymptotic results at hand. A usual practice is to extend the database by duplicating R times the original one and adding independent and identically distributed noises in each duplication. The following result gives then more insights in what happens in the finite sample case:

Theorem 2 *Under the same hypotheses as in theorem 1, and if every $N_i > 0$ remains fixed, then as R tends to infinity $\Phi(\cdot, W)$ tends to the best approximation of the estimated Bayesian solution, obtained by replacing densities by their kernel estimates.*

The proof given in appendix also gives conditions on the noise density for the estimate to be consistent. In particular, there is a close link between the noise density and the kernel function used (see section 5.). It seems thus more direct to build constructively the asymptotic limit, towards which the MQE solution will tend in the best case (*i.e.* if the absolute minimum is reached). And this leads us to kernel estimators of density.

5. Kernel estimators

One of the most interesting estimator of densities is known as the *kernel estimator*, sometimes abusively called *Parzen estimator*, as we shall see. It has not only nice consistency properties, but also can provide continuous estimates regardless of the number of patterns available in the learning set, which is of great practical interest, as opposed to histograms for instance. General statements about kernel estimators can be found in [23] [18] [27] [11] [19].

With the notation introduced in section 3., the kernel estimate of $p(x|\omega_j)$ takes the form:

$$\hat{p}(u|\omega_i) = \frac{1}{N_i} \sum_{x(n) \in A_i} \frac{1}{h(n;i)^d} K\left(\frac{u - x(n)}{h(n;i)}\right), \tag{10}$$

where $h(n;i)$ is strictly positive and $K(\cdot)$ is the kernel function. The choice of the kernel gives the estimator its basic finite sample properties; for instance, if $K(\cdot)$ is positive and twice differentiable, then so is $\hat{p}$.

If $h(n;i)$ depends only on N_i and not on n, then the estimator is said to have a fixed width, or to be a fixed kernel estimator, in short. This estimator was originally proposed by Parzen [24], and Cacoullos [4] extended it to the multichannel case. The suggestion of a variable width has been proposed independently by Wagner [29] and Breiman [3]. Thus the variable kernel estimator should not be called a Parzen estimator, for the sake of clarity.

Throughout this section, the reasoning is carried out for a fixed class ω_i, so that for conciseness index i may be dropped, being understood that statements exposed for $\mathcal{A}, h(n), N, \hat{p}(u)$ will be applied to $\mathcal{A}_i, h(n;i), N_i, \hat{p}(u|\omega_i)$, and so forth.

Moreover, it is convenient to decompose the width factor $h(n)$ into a global factor h and a local weighting factor $\eta(n)$, so that estimator (10) rewrites for any fixed class label i:

$$\hat{p}(u) = \frac{1}{N} \frac{1}{h^d} \sum_{n=1}^{N} \eta(n)^d K\left(\eta(n) \frac{u - x(n)}{h}\right). \tag{11}$$

Of course since this decomposition is not unique, one can arbitrarily impose in addition that $\prod_{n=1}^{N} \eta(n) = 1$.

5.1. Fixed kernel estimator

It is assumed here that $\eta(n) = 1, \forall n$. In the finite sample case, it has been proved by Rosenblatt [25] that kernel estimators of density are always biased, except for particular distributions.

On the other hand, they can be proved to be consistent. In fact, under the following conditions, it has been proved that the estimator $\hat{p}(u)$ is asymptotically unbiased:

$$K(u) > 0, \quad K(u) < \infty, \quad and \quad \int K(u)\, du = 1, \tag{12}$$

$$||u||^d K(u) \to 0 \quad as \quad ||u|| \to \infty, \tag{13}$$

$$h \to 0 \quad as \quad N \to \infty, \tag{14}$$

$$N\, h^d \to \infty \quad as \quad N \to \infty. \tag{15}$$

Conditions (12) to (14) ensure that $\hat{p}(u)$ converges to $p(u)$ at every continuity point of $p(u)$, but the convergence is in mean. With the additional condition (15) on the convergence speed of h, then the convergence is in quadratic mean. Actually similar results hold under slighlty less restrictive assumptions (*e.g.* the kernel may not be requested to be positive) [4] [29] [21].

The proof is easy to carry out when the kernel is twice differentiable and symmetric about the origin. Moreover in that case, it is possible to argue for a choice of an optimal width. In fact, after the change of variable $y = (u-x)/h$, bias and variance of estimator (11) can be expressed as:

$$B(u) = \int K(y)\, p(u-hy)dy - p(u), \tag{16}$$

$$V(u) = \frac{1}{N\, h^d} \int K^2(y)\, p(u-hy)dy - \frac{1}{N} \left[\int K(y)\, p(u-hy)dy\right]^2. \tag{17}$$

Now expand $p(u-hy)$ about u as:

$$p(u-hy) = p(u) + h\, \dot{p}(u)^T y + \frac{h^2}{2}\, y^T \ddot{p}(u)\, y + O(h^3), \tag{18}$$

where $\dot{p}$ is the gradient of p and $\ddot{p}$ the matrix of its second derivatives. Next using the fact that $K(\cdot)$ is symmetric about the origin, we obtain the asymptotic approximations:

$$B(u) = \frac{h^2}{2} \operatorname{Trace}\{\ddot{p}(u)\, V_K\} + O(h^4), \tag{19}$$

$$V(u) = \frac{1}{N\, h^d}\, \beta_K p(u) + O\left(\frac{1}{N\, h^{d-2}}\right). \tag{20}$$

where $\beta_K \stackrel{\text{def}}{=} \int K^2(u)\, du$ and $V_K \stackrel{\text{def}}{=} \int K(u)\, u\, u^T\, du$. The case where the kernel is isotropic is interesting, *i.e.* , when $K(\cdot)$ is a function only of the norm of its argument. Then $V_K = I_d$, and the bias reduces to the simple expression $B(u) = h^2\, \Delta p/2$, Δp denoting the Laplacian of p.

Clearly, as h decreases, the bias decreases but the variance increases. The trade-off is to minimize the integrated mean square error:

$$e(h) = \int e(u;h)\, du; \quad e(u;h) = B(u)^2 + V(u). \tag{21}$$

Then it is easy to see that this error reaches a unique minimum for a value h_o satisfying:

$$N\, h_o^{d+4} \int \Delta^2 p(u)\, du = d\, \beta_K. \tag{22}$$

Three conclusions can be drawn from here. First, we have an (asymptotically) optimal value for the width factor, provided Δp is given. In practice, the calculation of $\int \Delta^2 p$ requires the use of a rough estimator, based on (32) for instance. We shall go back to that in the next section.

Second, the minimum error obtained is

$$e(h_o) = \left(\frac{d}{4} + 1\right) \frac{1}{N\,h^d}\,\beta_K.$$

This minimal error is of order $O(N^{-1}h^{-d})$. In other words, since $h_o = O(N^{-1/(d+4)})$, $e(h_o) = O(N^{-4/(d+4)})$, and $B(u)$ and $V(u)$ are of same order.

Third, $e(h_o)$ is proportional to β_K. It is thus convenient to utilize kernel functions that have a small β_K.

Actually, one can even find what is the best kernel function to be used with this respect. Epanechnikov had early noticed this fact in the scalar case [13], resorting to standard tools from calculus of variations. But this extends to the multivariate case, if $K(\cdot)$ is isotropic: the positive kernel that minimizes β_K under the constraint of unit covariance is given by

$$K(u) = a - b\,u^T u, \quad \text{for } u^T u \leq a/b, \quad \text{and} \quad K(u) = 0 \,\text{elsewhere}. \tag{23}$$

Coefficients a and b are choosen so as to satisfy $\int K(u)\,du = 1$ and $\int K(u)\,u^T u\,du = d$:

$$K(u) = \frac{d+2}{2\,\gamma_d} \left(\frac{d+4}{d}\right)^{-\frac{d}{2}} \left[1 - \frac{d}{d+4}\,u^T u\right], \tag{24}$$

and γ_d is the volume of the the unit bowl in dimension d:

$$\gamma_d \stackrel{\text{def}}{=} \frac{\pi^{d/2}}{\Gamma(d/2+1)}. \tag{25}$$

The point is that the kernel obtained is of compact support, which is of practical interest: in order to compute the density at a point u, only patterns located in the neighborhood of u are necessary (in a bowl of radius $h\sqrt{1+4/d}$).

Though suboptimal from this point of view, the family of so-called *generalized Gaussian* kernels are of interest in certain cases [16]:

$$K_g(u) = B_g\,e^{-[A_g u^t u]^g}, \tag{26}$$

where coefficients A_g and B_g are to be determined in order to have a unit sum of the density, and a unit variance. The exact expressions of these coefficients are:

$$B_g = g\,\frac{a\,b^{d/2}}{c^{1+d/2}\,(\pi d)^{d/2}}, \quad A_g = \frac{b}{cd}, \tag{27}$$

$$\text{with } a = \Gamma(\frac{d}{2}),\; b = \Gamma(\frac{d+2}{2g}),\; c = \Gamma(\frac{d}{2g}). \tag{28}$$

If $g = 1$, the distribution is clearly Gaussian. One of the advantages in using such kernels, is that they have large tails, which allows making a non ambiguous decision in a larger domain in the input space.

5.2. Variable kernel estimators

For finite sample sizes, the fixed kernel estimate is not very satisfactory. In fact, one one hand isolated patterns are supposed to account for tails of the density, and should have a large width factor, whereas concentrated patterns that are supposed to produce a sharp peak should have a small width factor. Keeping the same width factor everywhere obviously impedes meeting those uncompatible requirements.

It is clear that the bias will be much better if the width factor is allowed to vary with location, especially for samples of reduced size. This has been noticed in [3] and proved in [22] for k-NN kernel estimators.

Denote $D_k(u)$ the distance between point u and its nearest k^{th} neighboring pattern. If $V_D(u)$ is the volume of the hypersphere of radius $D_k(u)$ centered at u, we must have as N tends to infinity:

$$\frac{k}{N} \approx p(u)\, V_D(u), \tag{29}$$

by definition of a density. This yields the classical result that the density may be roughly approximated by:

$$\hat{p}(u) \approx \frac{k}{N}\,\frac{1}{V_D(u)}, \tag{30}$$

where $V_D(u) = \gamma_d\, D_k(u)^d$, with γ_d as defined in (25).

This classical approach cannot be used directly since the estimate obtained is not continuous, and integrates to infinity. A simple way to fix this problem is to use $D_k(u)$ to choose the local weighting factor $\eta(n)$ in (11):

$$\eta(n) = \frac{h}{D_k(n)}, \tag{31}$$

$$h = \sqrt[N]{\prod_{n=1}^{N} D_k(x(n))}. \tag{32}$$

Note that if we use a uniform kernel (constant value in the unit bowl and zero elsewhere), then we get the k-NN estimator (30).

Mack and Rosenblatt have analyzed the asymptotic local bias and variance of such an estimator. They have found that the optimal value for integer k should increase with N as N^b, $b = 4/(d+4)$ [22]. In practice, one should also pay attention to the fact that k should be large enough to avoid a null value of $D_k(x(n))$ for some n. Another way to avoid null values of $D_k(x(n))$ is to clip them below. In any case, if we assume a law of the form $k = a\,N^b$, the constant factor a remains very difficult to find. This is the same problem as for h in the fixed kernel estimator.

Even if the latter estimator is apparently of better use, it may still not integrate to one, and is not differentiable [22]. For this reason, it has been adopted in *Packlib* [5] a more sophisticated approach. The local weighting factor $\eta(n)$ is estimated by minimizing the local mean square error $e(u; h)$ defined in (21); this idea is (comparatively) quite recent, for it has been first advanced twelve years ago in [1]. Here, this error of course also depends on $\eta(n)$, and our goal is now to compute its bias and variance components.

Start as in the fixed kernel case, and write the mean of estimator (11):

$$E\hat{p}(u) = \int \frac{\eta(x)^d}{h^d} K\left(\eta(x)\,\frac{u-x}{h}\right) p(x)\,dx \tag{33}$$

The expectation is eventually the limit of the finite sum as N tends to infinity. For convenience, the —somewhat abusive— notation $\eta(x)$ has been assumed, being understood that $\eta(x(n)) = \eta(n)$. Now perform the change of variable as before, $y = (u-x)/h$, and obtain:

$$E\hat{p}(u) = \int \eta(u-hy)^d\, K(y\,\eta(u-hy))\, p(u-hy)\,dy. \tag{34}$$

The difference is that now 3 terms must be expanded in Taylor series, compared to a single one in the fixed kernel case. The expansions are written in a similar manner as in (18). After a number of rather heavy manipulations, and taking into account the symmetry of the kernel function, we obtain:

$$\begin{aligned} B(u) \;&=\; \frac{h^2}{\eta^2(u)}\,\mathrm{Trace}\left[V_K\left(\frac{\ddot{p}(u)}{2} - 2\,\frac{\dot{\eta}(u)\,\dot{p}(u)^T}{\eta(u)} - \frac{\ddot{\eta}(u)\,p(u)}{\eta(u)}\right.\right. \\ &\quad \left.\left. +3\,\frac{\dot{\eta}(u)\dot{\eta}(u)^T}{\eta^2(u)}\,p(u)\right)\right] + O(h^4). \end{aligned} \tag{35}$$

In the scalar case ($d = 1$), this formula reduces to that obtained in [1]. As Abramson pointed out, if $\eta(u) = p(u)^{1/2}$, this bias reduces to $O(h^4)$. On the other hand, the expansion of the variance is very simple, since it is sufficient (as in the fixed kernel case) to go up to order zero:

$$V(u) = \frac{1}{N\,h^d}\,\beta_K\,\eta(u)\,p(u) + O\left(\frac{1}{N\,h^{d-2}}\right). \tag{36}$$

The most surprising fact is that with an appropriate choice of $\eta(u)$, the bias is cancelled, leaving total freedom to choose h in order to reduce arbitrarily the variance (*i.e.* h large).

There are some practical comments to make, that limit those conclusions. If $\eta(u) = p(u)^{1/2}$ is chosen, then it may happen that $\eta(u)$ be null, which is forbidden by our assumptions. If necessary, $\eta(u)$ may have to be clipped below. Next, since h needs to be small for the expansion of the bias to be valid, an arbitrarily large value of h would not be acceptable.

The complete practical algorithm is described in section 7., and overcomes these difficulties. Other criteria than the mean square error can be used, and in particular, the final classification error [8].

6. Suboptimality by clustering

There are two reasons for building suboptimal solutions based on clustering. Imagine our classifier is intended to be implemented in a cheap product, and that there are strong hardware limitations. Then estimator (11) cannot be used directly because all patterns $x(n)$ need to be stored. So it is relevant in that case to design a classifier that would use the available resources in the very best way, regardless of the learning complexity [8]. The other reason is that if it is wished to still use estimator (11), then a rough (pilot) estimator of the density is necessary to determine h. The estimator presently proposed can be used for this purpose.

In this section, we consider that data in each class ω_i of the learning set $\mathcal{A}(N)$ have been clustered into Q_i disjoint groups $G_{q;i}, 1 \leq q \leq Q_i$. Denote $N(q;i)$ the number of patterns in group $G_{q;i}$. We have $\sum_{q=1}^{Q_i} N(q;i) = N_i$. Then from the Bayes rule:

$$p(u|\omega_i) = \frac{1}{P_i} \sum_{q=1}^{Q_i} P(G_{q;i})\, p(u|G_{q;i}). \tag{37}$$

This shows first that the density may change after vector quantization, because of the presence of weigths $P(G_{q;i})$. Next, this relation suggests the following reconstruction formula, if $P(G_{q;i})$ is estimated by the ratio $N(q;i)/N$, and P_i by $\hat{P}_i = N_i/N$:

$$\tilde{p}_I(u|\omega_i) = \frac{1}{N_i} \sum_{q=1}^{Q_i} \frac{N(q;i)}{\sigma(q;i)^d}\, K\left(\frac{u - C[q;i]}{\sigma(q;i)}\right), \tag{38}$$

where a single width factor $\sigma(q;i)$ has been used within each cluster. This relation can equivalently be obtained by replacing $x(n)$ by the centroid $C[q;i]$ of its group in (10). In this approach, all clusters are spherical. This may be a problem because a large number of spherical clusters may be required to approximate data containing anisotropic clusters.

To palliate this limitation, another more accurate reconstruction procedure involves a positive definite matrix $L[q;i]$ that accounts for clusters shape:

$$\tilde{p}_A(u|\omega_i) \quad = \quad \frac{1}{N_i} \sum_{q=1}^{Q_i} \frac{N(q;i)}{h(q;i)^d}\, K\left(\frac{L[q;i]^{-1}(u - C[q;i])}{h(q;i)}\right). \tag{39}$$

It remains to estimate centroids $C[q;i]$, shape factors $L[q;i]$, and width factors $\sigma(q;i)$ or $h(q;i)$. We describe below one reasonable solution. Assume the clusters are sufficiently well separated so that the kernel tails of neighboring clusters vanish. Then, with the description above, the density estimate within a cluster reduces to a single mode. Yet, in reconstructions (38) and (39), only moments of order 1 and 2 are used, so that the density within a cluster may be approximated by a Gaussian density. In other words, $K(\cdot)$ can be assumed to be a radial Gaussian kernel for this calculation (but only that one):

$$K(u) = (2\pi)^{-d/2}\, exp\{-||u||^2/2\}.$$

With this approximation, maximum likelihood estimates can be easily computed. If $\tilde{p}_{I}(u|\omega_i)$ is maximized with respect to $C[q;i]$ and $\sigma(q;i)$, we obtain:

$$C[q;i] = \frac{1}{N(q;i)} \sum_{x(n)\in G_{q;i}} x(n), \tag{40}$$

$$\sigma(q;i)^2 = \frac{1}{d}\,\frac{1}{N(q;i)} \sum_{x(n)\in G_{q;i}} ||x(n) - C[q;i]||^2. \tag{41}$$

Next, if $\tilde{p}_{A}(u|\omega_i)$ is maximized with respect to $C[q;i]$, $h(q;i)$ and $L[q;i]$, we obtain:

$$L[q;i]\, L[q;i]^T = A[q;i], \quad \text{with} \tag{42}$$

$$A[q;i] = \frac{1}{N(q;i)} \sum_{x(n)\in G_{q;i}} (x(n) - C[q;i])(x(n) - C[q;i])^T, \tag{43}$$

$$h(q;i)^d = \det L[q;i]. \tag{44}$$

Thus, $L[q;i]$ is any positive square root of $A[q;i]$, for instance its lower triangular Cholesky factor. Of course, this estimate is biased. To remove the bias, one can replace $N(q;i)$ by $N(q;i) - 1$, as usual.

However, it is clear that these solutions are not the best possible, neither with respect to criterion $e(u;h)$ defined in (21), nor with respect to the misclassification rate. Since the final goal is actually classification, a criterion measuring deviations from the true densities, like $e(u;h)$, is not the most appropriate, especially if memory resources are strongly limited.

Thus, it has been proposed in [8] to find the best parameter set, $\{Q_i, C[q;i], \sigma(q;i), h(q;i), L[q;i]\}$, in the Bayes sense. The best solution obtained may –or may not– yield good density estimates, it does not matter, but it will lead to the best classification rate. Other approaches exist if the number of clusters is large, and their size small [28].

7. Computational aspects

7.1. Computation of the Laplacian

There are practical obstacles in computing the optimal value of the kernel width h. One could think of computing it by using (22), where the Laplacian of the true $p(u)$ is replaced by the one of a rough estimate $\hat{p}_R(x)$, yielding:

$$\widehat{\Delta\, p}(u) \approx \Delta\, \hat{p}_R(u) = \frac{1}{N\,\hat{h}^3} \sum_{n=1}^{N} \hat{\eta}(n)^{d+2}\, \Delta K\left(\hat{\eta}(n)\, \frac{u - x(n)}{\hat{h}}\right). \tag{45}$$

Estimated widths $\hat{h}$ and $\hat{\eta}$ can be those given by (32) and (31) for instance. But the integration is made difficult because of the squaring of the sum, that introduces cross-terms.

A "brute force" approximation is then to replace the integration by a discrete sum over the available patterns $x(n)$:

$$\int \Delta^2 p(u)\, du \approx \frac{1}{N} \sum_{n=1}^{N} \Delta^2 \hat{p}_R(x(n)). \tag{46}$$

The obtained value (though acceptable) is likely to be too large [10].

Another alternative is to use the estimate provided by clustering in section 6. The rough estimate is of the form:

$$\hat{p}_R(u|\omega_i) = \sum_{q=1}^{Q_i} p(u|G_{q,i}) \tag{47}$$

$$= \sum_{q=1}^{Q_i} [\det(2\pi\, A[q;i])]^{-1/2}\, exp\{-\frac{1}{2}(u - C[q;i])^T A[q;i]^{-1}(u - C[q;i])\}.$$

Now, since the Gaussian family enjoys a reproductive property, cross products yield again a Gaussian density up to a multiplicative term, and are easy to integrate. As a consequence, we have access to an analytic expression of $\int \Delta^2 \hat{p}_R(u)\, du$. Approximating the density $p(u)$ by a Gaussian mixture where the parameters are estimated suboptimally might be considered very crude, but it is not. Of course one could optimize these parameters further [8], but it would be superflous for the only purpose of computing $\int \Delta^2 p(u)\, du$.

7.2. Description of the algorithm

The kernel choosen is isotropic (radial function), and can be either the Epanechnikov kernel (24), or a generalized Gaussian kernel (26). The algorithm proposed to compute the estimate (11) is two-pass (Rough-Refined Estimator):

1. A rough estimate $\hat{p}_R(x)$ is obtained at any point $x(n)$ of the learning set by using the kernel k-NN estimator (11) with $\hat{\eta}_R(x)$ and $\hat{h}_R$ defined by (31), (32).

2. Next, this variable kernel estimate is refined by assuming the new value of $\hat{\eta}(n)$:

$$\hat{\eta}(n) = [\hat{p}_R(x_n)]^{1/2} \left[\prod_{n=1}^{N} \hat{p}_R(x_n) \right]^{-1/2}. \tag{48}$$

3. Optionally, a new value of $\hat{h}$ may be refined in accordance with section 7.1.

Performances are evaluated by using the ALOO procedure briefly described in section 3.2.

8. Suboptimality by dimension partitioning

In order to estimate a density in dimension d with a given accuracy, there is a minimum number N_{min} of samples required. This number is unfortunately an exponential function of the dimension. Of course, some estimators perform better than others, but the tendency as d increases is the same for all of them. As a consequence, one can expect that it will be very difficult to estimate a density in large dimension, because samples will be too small.

It is worth knowing what is a small sample size, and what is a large dimension. Silverman reported the value of N_{min} as a function of d. Based on extensive simulations, he found the value of N that gave a relative error on a Gaussian density of 10%, when using a kernel estimator with Gaussian kernel [27].

An affine approximation of $\log N$ would gives the following result [9]:

$$\log_{10} N_{min} > 0.6\,(d - \frac{1}{4}). \tag{49}$$

Let's just give an example. If $d = 10$, this gives $N_{min} > 708,000$. This motivates strongly the reduction of the dimension. However, there are cases where one cannot project the data on a smaller-dimensional subspace without loosing significant information.

In this section, we propose another approach based on dimension partitioning. Assume that a density $p(u)$ defined on the space $\mathcal{E} = I\!R^d$ can be approximated by a product of densities as:

$$p(u) \approx p_1(u_1)\, p_2(u_2), \tag{50}$$

where $u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$. Then densities p_1 and p_2 can be much more easily estimated since they share the same sample size, but need to be estimated in reduced dimension. If this is not enough, the procedure can be iterated further.

But the decomposition (50) means that the d-dimensional random variable x has been splitted into two *statistically independent* random variables x_1 and x_2. This problem, that we can refer to as the Independent Subspace Analysis (ISA), was addressed in [7], and an algorithm has been proposed there to construct variables u_1 and u_2. We consider this is a major area of research for future years, both in data analysis and numerical analysis.

References

[1] I.S. ABRAMSON, "On bandwidth variation in kernel estimates - a square root law", *The Annals of Statistics*, vol. 10, no. 4, pp. 1217–1223, 1982.

[2] L. BREIMAN, J.H. FRIEDMAN et al, *Classification and Regression Trees*, Wadsworth, 1984.

[3] L. BREIMAN, W. MEISEL, E. PURCELL, “Variable kernel estimates of multivariate densities”, *Technometrics*, vol. 19, no. 2, pp. 135–144, May 1977.

[4] T. CACOULLOS, “Estimation of a multivariate density”, *Annals of Inst. Stat. Math.*, vol. 18, pp. 178–189, 1966.

[5] Y. CHENEVAL, “Packlib, an interactive environment to develop modular software for data processing”, in *Conference IWANN*, Mira Cabestany Prieto, Ed., Malaga, Spain, Jun 7–9 1995, Springer-Verlag, Lecture Notes in Computer Sciences.

[6] P. COMON, “Classification supervisee par reseaux multicouches”, *Traitement du Signal*, vol. 8, no. 6, pp. 387–407, Dec. 1991.

[7] P. COMON, “Independent Component Analysis, a new concept ?”, *Signal Processing, Elsevier*, vol. 36, no. 3, pp. 287–314, Apr. 1994, Special issue on Higher-Order Statistics.

[8] P. COMON, Y. CHENEVAL, “Bayesian supervised classification: an approach with variable kernel estimators”, in *Conference IWANN*, Mira Cabestany Prieto, Ed., Malaga, Spain, Jun 7–9 1995, Springer-Verlag, Lecture Notes in Computer Sciences.

[9] P. COMON, C. JUTTEN et al., “ELENA deliverable R1-A-P, Axis A: Theory”, Esprit Basic Research Project 6891, CEC, June 1993.

[10] P. COMON, J. L. VOZ, M. VERLEYSEN, “Estimation of performance bounds in supervised classification”, in *ESANN-European Symposium on Artificial Neural Networks*, M. Verleysen, Ed., 45 rue Masui, B-1210 Brussels, Belgium, April 20-22 1994, pp. 37–42, D facto Publ.

[11] M. DELACROIX, *Histogrammes et Estimateurs de densite*, P.U.F., 1983.

[12] R.O. DUDA, P.E. HART, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[13] V.A. EPANECHNIKOV, “Non-parametric estimation of a multivariate probability density”, *Theory Proba. Appl*, vol. 14, pp. 153–158, 1969.

[14] K. FUKUNAGA, *Introduction to Statistical Pattern Recognition*, Academic Press, 1972.

[15] K. FUKUNAGA, R.R. HAYES, “Estimation of classifier performance”, *IEEE Trans. PAMI*, vol. 11, no. 10, pp. 1087–1101, Oct. 1989.

[16] K. FUKUNAGA, D.M. HUMMELS, “Bayes error estimation using Parzen and k-NN procedures”, *IEEE Trans. PAMI*, vol. 9, no. 5, pp. 634–643, Sept. 1987.

[17] D.J. HAND, *Discrimination and Classification*, Wiley, 1981.

[18] D.J. HAND, *Kernel Discriminant Analysis*, RSP press, 1982.

[19] W. HÄRDLE, *Smoothing Techniques, with implementation in S*, Springer-Verlag, 1990.

[20] N. L. JOHNSON, S. KOTZ, *Distributions in statistics: Discrete Distributions*, Wiley, 1969.

[21] R. J. KARUNAMUNI, K. L. MEHRA, "Optimal convergence properties of kernel density estimators without differentiability conditions", *Annals Inst. Statist. Math.*, vol. 43, pp. 327–346, 1991.

[22] Y. P. MACK, M. ROSENBLATT, "Multivariate k-Nearest Neighbor density estimates", *Jour. Multivariate Analysis*, vol. 9, pp. 1–15, 1979.

[23] J.S. MARITZ, *Distribution-Free Statistical Methods*, Chapman and Hall, 1981.

[24] E. PARZEN, "On the estimation of a probability density function and the mode", *Ann. Math. Stat.*, vol. 33, pp. 1065–1076, 1962.

[25] M. ROSENBLATT, "Remarks on some non parametric estimates of a density", *Ann. Math. Stat.*, vol. 27, pp. 832–837, 1956.

[26] D. W. RUCK, S. K. ROGERS, "The multilayer perceptron as an approximation to Bayes optimal discriminant function", *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, Dec. 1990.

[27] B.W. SILVERMAN, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986.

[28] J. L. VOZ, M. VERLEYSEN, P. THISSEN, "A practical view of sub-optimal Bayesian classification with radial Gaussian kernels", in *Conference IWANN*, Mira Cabestany Prieto, Ed., Malaga, Spain, Jun 7–9 1995, Springer-Verlag, Lecture Notes in Computer Sciences.

[29] T. J. WAGNER, "Nonparametric estimates of probability densities", *IEEE Trans. on Inf. Theory*, vol. 21, no. 4, pp. 438–440, July 1975.

Appendix

Proof of theorem 2

The proof given here is a particular case of another one already published in French in [6], where general losses were considered.

Proof. Denote ε the MQE criterion, for finite N and R:

$$\varepsilon(N,R) = \sum_{k=1}^{K} \frac{N_k}{N} \frac{1}{N_k} \sum_{x(n)\in\omega_k} \frac{1}{R} \sum_{r=1}^{R} ||z_{k(n)} - \Phi(x(n) + z(n,r), W)||^2, \quad (51)$$

where the $w(n,r)$'s denote additive noises drawn from a given density $p_z(u)$. Now let:

$$\begin{aligned}
\hat{P}_k &= \frac{N_k}{N} && (52)\\
\xi_k(u) &= ||z_{k(n)} - \Phi(u,W)||^2 \quad \text{for } x(n) \in \omega_k. && (53)
\end{aligned}$$

This is possible since the output $y(n)$ depends only on k. Assume every N_i is non zero and let R tend to infinity. We get:

$$\varepsilon(N,\infty) = \sum_{k=1}^{K} \hat{P}_k \int p_z(u) \frac{1}{N_k} \sum_{x(n)\in\omega_k} \xi_k(x(n)+u)\, du. \tag{54}$$

Make the change of variable $v = x(n) + u$ and define

$$\hat{p}(v|\omega_k) = \frac{1}{N_k} \sum_{x(n)\in\omega_k} p_z(v - x(n)). \tag{55}$$

It can be obtained then:

$$\varepsilon(N,\infty) = \sum_{k=1}^{K} \hat{P}_k \int \hat{p}(v|\omega_k)\, \xi_k(v)\, dv. \tag{56}$$

Define next:

$$\begin{aligned}
\hat{p}(v) &= \sum_{k=1}^{K} \hat{P}_k\, \hat{p}(v|\omega_k), && (57)\\
\hat{g}_k(v) &= \frac{\hat{P}_k\, \hat{p}(v|\omega_k)}{\hat{p}(v)}. && (58)
\end{aligned}$$

Now the error can be expressed as:

$$\varepsilon(N,\infty) = \int \hat{p}(v)\, ||\Phi(v,W)||^2\, dv - 2 \int \sum_{k=1}^{K} \hat{g}_k(v)\, \Phi_k(W,v)\, dv + \varepsilon_1, \tag{59}$$

where ε_1 is independent of the Φ_k's. A short manipulation finally leads to:

$$\varepsilon(N,\infty) = \int \hat{p}(v)\, ||\Phi(v,W) - \hat{g}(v)||^2\, dv + \varepsilon_2, \tag{60}$$

where ε_2 is independent of vector Φ, and $\hat{g}$ is the vector with components $\hat{g}_k$. This last result shows that the mapping $\Phi(\cdot, W)$ obtained is the one closest to $\hat{g}(v)$. Yet, this is an estimate of the Bayesian discriminating functions $g_k(v) = P_k\, p(v|\omega_k)/p(v)$. In other words, if the family of functions $\Phi(\cdot, W)$ is sufficiently large, the largest $\Phi_k(W; v)$ will be reached for the same k as the largest $\hat{g}_k(v)$, yielding the same decision. $\square$

ESANN'95, 3rd Europeean Symposium on Artificial Neural Networks,
Brussels, April 19-21, 1995, pp 111–128.
D facto Publications, 45 rue Masui, B-1210 Brussels.