IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Statistical Learning Theory Approach To Trade Off Between Samples and Features in Classification Problems

JOÃO ISMAEL PINHEIRO[1] DANIEL SADOC MENASCHÉ[1] (Member, IEEE), RODRIGO MELLO[2], HUGO CARVALHO[1]

[1]Federal University of Rio de Janeiro (UFRJ) (e-mail: sadoc@dcc.ufrj.br, cabrallima@ufrj.br)
[3]USP (e-mail:)

Corresponding author: João Ismael Pinheiro (e-mail: jismael@im.ufrj.br).

This work was supported in part by THANES, a joint project between Brazil (FAPERJ) and France (INRIA), and by grants from CAPES and CNPq.

ABSTRACT

In a binary classification problem, the main classifier quality indicator is its expected error rate, which ideally should be as small as possible. In this article we analysed the joint effect of changes in: dataset cardinality, number of active features, dictionary choice, etc. on a classifier's expected error rate. One of our main concerns was to investigate what happens as the dataset cardinality grows indefinitely.

For that purpose, we adopted an experimental approach based on simulated data. Our results show that increasing dataset cardinality can mitigate an eventual damage caused by lacking some important features. However, it also indicates that the expected error rate decrease is very slow as dataset cardinality grows. Consequently, a possible trade-off between cardinality and dimensionality by itself seems to be, at least, controversial in this context.

We also investigated the interaction between this cardinality/dimensionality issue and the role of an adequate dictionary choice. Here, our simulation results show that, in order to guarantee the success of a classifier's performance, making an appropriate dictionary choice could be even more important than adding new relevant features to the current dataset. As a by-product of that analysis, we proved some theoretical results about the expected error rate evolution as dataset cardinality grows towards infinity.

We also proved that, if the VC dimension of the chosen dictionary is proportional to the feature space dimensionality, then both random variables: the stochastic error and the estimation error converge in probability to zero, as cardinality tends to infinity. Our simulations show that, as cardinality grows, the empirical cumulative distributions of both stochastic error and estimation error become closer and closer to a step function at zero, in all the situations we investigated. Clearly, this evidence confirms we do have convergence in probability to zero, for both errors.

INDEX TERMS Statistical learning theory, classification, machine learning

I. INTRODUCTION

Let us consider a classification problem with only two classes, which will be represented by the labels -1 and +1.

The problem can be formulated as follows:

P is the joint probability distribution of the random vector (X, Y), where $X \in R^d$ and $Y \in \{-1, +1\}$.

Of course, there are other aspects that could also be introduced, such as: choosing a prior distribution for Y, introducing a loss function, etc. But, for simplicity, let us assume equal priors and a 0 or 1 loss function.

We will begin with some definitions: A **classifier** is a measurable function h: $R^d \to \{-1, +1\}$. For each classifier h, let be $L(h) = P(Y \neq h(X))$, i.e., L(h) is the **theoretical error rate** (or, the **theoretical loss**) for classifier h. A **dictionary** (also known as **search space bias**) H is a family of classifiers (e.g., linear classifiers, quadratic classifiers,...) in R^d, from which we will select one specific classifier that supposedly will be the best possible solution for our classification problem.[1], [2] ou [3], [4], [5], [6], [7], [8], [9]

II. STATISTICAL LEARNING THEORY PRIMER - CLASSIFICATION PROBLEM

A. HOW TO EVALUATE A CLASSIFIER'S EFFICIENCY

Under these conditions, what should be our goal while facing a binary classification problem? Common sense indicates that we should look for a classifier with the lowest possible error rate.

Suppose we have a real dataset D on a given topic. In that case, what is the error rate that really matters? Let us assume the available data are used to calibrate a classifier whose empirical error rate is as small as possible when applied to dataset D. Then, of course, the best way to judge this classifier's efficiency would be predicting its empirical error rate whenever applied to another independent dataset D'. If we are lucky enough to get a dataset with many observations, the usual solution would be randomly dividing the available dataset into train and test. This would avoid underestimating the actual error rate, by using the same data to calibrate the classifier and also to evaluate its effectiveness. In this context, it is important to distinguish between the empirical and the theoretical error rates associated with a given classifier: the empirical rate is the train rate (which tends to underestimate the real error rate); while the test rate provides a more reliable estimate of its expected error rate.

Also, if it is not possible to get a dataset with many observations, then a resampling technique (such as cross-validation, bootstrap, etc.) could be used to avoid such bias in estimating the rate of classification errors.

Now, what if we work with simulated data coming from a known probabilistic model? In that case, for a given classifier, it is always possible to use probability theory in order to calculate (eventually by numerical methods) its theoretical error rate, without having to evaluate its effectiveness through test data.

Since this is a methodological paper, we chose to work with simulated data, which greatly facilitates the task of assessing each classifier's effectiveness in an unbiased way. Fortunately, conclusions based on this type of approach are also applicable to concrete situations in which we work with real data.

On the other hand, when we face a binary classification problem, what are our "control buttons"? Among them, we can mention:

- Dataset cardinality n, i.e., the number of available observations.
- The problem dimensionality d, i.e., the number of available features.
- Each feature discrimination power, either alone or in the presence of the others features.
- Dictionary H, from which we will pick our classifier.

It would then be worth asking how a particular combination choice of such "control buttons" will affect the expected rate of classification errors in our mathematical model.

B. LOSS MEASUREMENT PRECISION BEHAVIOUR, AS CARDINALITY GROWS

Another aspect that should not be overlooked is how precisely we can estimate the error rate of our classification procedure. And, of course, it is also interesting to investigate how each of the previously mentioned "control buttons" affects this precision. This is what we will discuss now. In particular, we will investigate what happens to the error rate estimation precision, as the dataset cardinality n tends to infinity.

For this purpose, we will first need some more definitions:

The **VC dimension** of dictionary H is the maximum number of points in R^d that can be arbitrarily classified (i.e., totally shattered) by the classifiers in H.

So, from now on, let H be a specific dictionary.

Also, suppose we have a dataset D, assumed to be representative of the phenomenon under study, and formed by the pairs (x_i, y_i), i = 1,2,. . . ,n, where $x_i \in R^d$ and $y_i \in \{-1, +1\}$, for all i = 1,2,. . . ,n.

Then, for each classifier h in H, we can calculate its **empirical error rate** $\hat{L}(h)$, i.e., its proportion of misclassified observations in dataset D:

$$\hat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} 1_{(y_i \neq h(x_i))}$$

Let $\hat{h}^{(D)}$ be the **best empirical classifier** in dictionary H, i.e., the classifier in H whose empirical error rate is minimum.

Under these conditions we can define three error rates:

- $\min_{h\in H} L(h)$ = the smallest achievable theoretical error rate in H
- $L(\hat{h}^{(D)})$ = the theoretical error rate of classifier $\hat{h}^{(D)}$
- $\hat{L}(\hat{h}^{(D)})$ = the empirical error rate of classifier $\hat{h}^{(D)}$

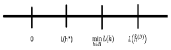


Figure 1: The Losses

It is a known fact that these losses necessarily obey an ordering (See Figure 1):

$$0 \leq L(h_*) \leq \min_{h\in H} L(h) \leq L(\hat{h}^{(D)}),$$

where $L(h_*)$ is the Bayes error rate (also known as the Bayes risk), i.e., the overall minimum achievable theoretical loss for this particular classification problem.

In this context we can also define three non-negative error quantities:

Approximation error: $\min_{h\in H} L(h) - L(h_*)$
Stochastic error: $\Delta L_1 = L(\hat{h}^{(D)}) - \min_{h\in H} L(h)$
Estimation error of $L(\hat{h}^{(D)})$: $\Delta L_2 = |L(\hat{h}^{(D)}) - \hat{L}(\hat{h}^{(D)})|$

While the approximation error is a constant, both ΔL_1 and ΔL_2 are random variables, since they depend on dataset D.

Putting together two results from Reference (A), namely "Control of the stochastic error" (Theorem 9.1) and "Sauer's Lemma" (Proposition 9.6), we got upper bounds for both ΔL_1 and ΔL_2. Under suitable conditions, these upper bounds converge to zero, as the dataset cardinality n goes to infinity. Also, since ΔL_1 and ΔL_2 are both random variables, the inequalities involving such bounding only prevail with a probability, which grows to 1 as n grows indefinitely. Based on these properties, we prove in Appendix A that, if the VC dimension d_H of dictionary H is proportional to d, the problem dimensionality, then both ΔL_1 and ΔL_2 converge in probability to zero, as n grows towards infinity.

C. A COMPUTATIONAL EXPERIMENT

In order to further investigate and illustrate this property, we have created a computational experiment with some simple examples where the theoretical probability distribution P is known. Then, in such cases it is possible:

- to compute exactly all theoretical error rates;
- to estimate the empirical error rates by simulation.

Furthermore, we considered only two simple kinds of dictionary boundaries, namely hyper-planes and hyper-rectangles. For both of them, the VC dimension is proportional to the feature space dimension, i.e., $d_H = d+1$, for linear classifiers and $d_H = 2d$, for hyper-rectangles. One of the goals of that investigation was to evaluate how tight or how loose are both inequalities in (*). For each one of six probability models (LIN1, RECT1, LIN2, RECT2, LIN3, RECT3), the idea was to observe the evolution of both ΔL_1 and ΔL_2 probability distributions as n grows towards infinity. For this purpose, we analysed their behaviour, by simulation, for three dataset sizes: 100, 1000, 10000. For each pair (model, dataset size), we generated 100 datasets by simulation. This strategy enabled us to build empirical cdf's for ΔL_1 (and for ΔL_2) corresponding to the 3 chosen dataset sizes: 100, 1000 and 10000. And by comparing these three situations we were able to experimentally check whether there is in fact convergence to zero in probability, as n grows to infinity.

Given a probabilistic model M, a specific dictionary H, and a dataset size n, we developed a program with the following common structure:

(a) Create a function ert.fn that, given a classifier h in H, calculates its theoretical error rate using probability theory according to model M.

(b) Using that function ert.fn, find the classifier in H whose theoretical error rate is minimum. In other words, in such a way compute $\min_{h\in H} L(h)$.

(c) Then create a function ere.fn that, for each classifier h in H and for each specific dataset (x_i, y_i), i = 1,2,...,n, calculates its empirical error rate by classifying the points according to that rule and counting the misclassified observations.

(d) Enter a loop with, say, m iterations. In each iteration:

  - A dataset D is generated by simulation, according to model M.
  - For each parameter defining a specific classifier in H, create a grid of possible values. (Note that a classifier h in H is defined through the specification of some parameters.)
  - Combining these grids, test several possible classifiers in H, using function ere.fn to compute their specific empirical error rates. This procedure enables to find the classifier $\hat{h}^{(D)}$ in dictionary H that minimizes the empirical error rate corresponding to dataset D. This minimum is what we call $\hat{L}(\hat{h}^{(D)})$.
  - Using function ert.fn, also compute $L(\hat{h}^{(D)})$.
  - Then ΔL_1 and ΔL_2 are calculated for dataset D.

(e) This procedure provides m values for ΔL_1 and m values for ΔL_2, and those values are used to obtain their empirical cumulative distribution functions (ecdf's), as well as location and dispersion measures.

D. THE THEORETICAL EXAMPLES

Tables 1 and 2 summarize some general information about the six examples we built. In all of them we used equal priors. Further information on the six examples can also be found in Appendix B.

Table 1: Summarizing the six examples

Example	Conditional Distributions of X given Y
LIN1	Univariate Gaussian versus Univariate Gaussian
RECT1	Univariate Gaussian versus Mixture of two Univariate Gaussians
LIN2	Bivariate Gaussian versus Bivariate Gaussian
RECT2	Bivariate Gaussian versus Mixture of four Bivariate Gaussians
LIN3	Trivariate Gaussian versus Trivariate Gaussian
RECT3	Trivariate Gaussian versus Mixture of six Trivariate Gaussians

Table 2: More details on the six examples

Example	d	Boundary	$\min_H L(h)$	Best theor. boundary
LIN1	1	1 cutpoint	0.159	c = 0
RECT1	1	2 cutpoints	0.397	(-1.09;+1.09)
LIN2	2	Line	0.240	$x_1 = x_2$
RECT2	2	Rectangle	0.418	$(-1.34; +1.34)^2$
LIN3	3	Plane	0.0416	$x_1 + x_2 + x_3 = 0$
RECT3	3	Parallelepiped	0.280	$(-1.73; +1.73)^3$

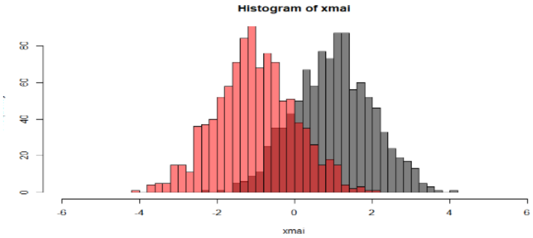


Figure 2: LIN1 - A histogram

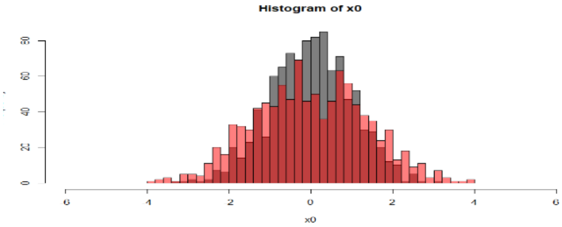


Figure 3: RECT1 - A histogram

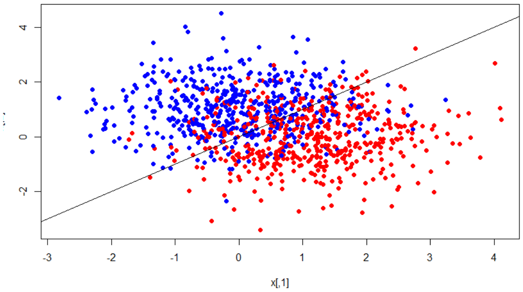


Figure 4: LIN2 - Scatterplot

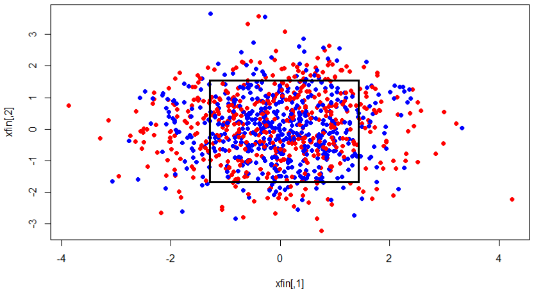


Figure 5: RECT2 - Scatterplot

According to the description of the computational experiment in II-C, for both ΔL_1 and ΔL_2, for all six examples, we built:

- Simultaneous graphs of their Empirical cumulative distribution function (ecdf), for three dataset sizes (namely, n = 100, n = 1000 and n = 10000).
- Tables with their Centrality and Dispersion measure estimates

All these results are in in Appendix D , and they seem to confirm that, for both ΔL_1 and ΔL_2, as the dataset size n grows towards infinity:

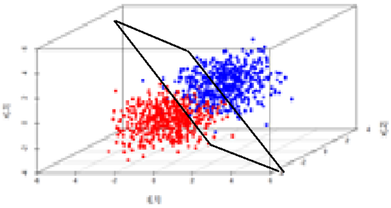


Figure 6: LIN3 - 3D scatterplot

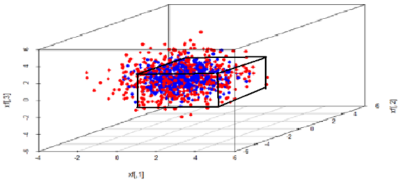


Figure 7: RECT3 - 3D scatterplot

- Their ecdf's become closer and closer to a step function at zero, which happens to be the cdf of a degenerate random variable equal to zero with probability 1.
- Their location (median) and dispersion (interquartile range) measures also tend to zero.

Notice that this behaviour is always present regardless of d, the feature space dimensionality, and for either linear or rectangular classifiers. Also, the same statistical tables enable us to evaluate how loose/how tight are the upper bounds we used to prove the convergence result in Appendix A.

E. SOME COMMENTS ON METHODOLOGY

1) Probabilistic models characteristics

Among the six examples we used in this article, in two of them x is one-dimensional (LIN1 and RECT1), in two of them x is two-dimensional (LIN2 and RECT2), and in two of them x is three-dimensional (LIN3 and RECT3). And for each pair of models with the same dimensionality of the x-space, in one of them the dictionary is composed by linear classifiers, while in the other one it is composed by rectangular classifiers. It is worth mentioning that each example was conceived in such a way there is a natural match between the probabilistic model and the choice of the particular dictionary.

2) Theoretical error rate calculation and minimization

In most of our six examples, the features joint probability distribution is a mixture of multinormals whose covariance matrix is always the identity matrix dxd. This fact implies conditional independence among the features, given a specific multinormal distribution. That property was important to simplify the computation of the theoretical error rates in Section 3. On the other hand, by construction, symmetry is present in all our examples. That means the theoretically minimizing classifier is also necessarily symmetric. This knowledge also helped simplifying the calculation of what we called $\min_{h \in H} L(h)$.

3) Empirical error rate minimization

In all our examples, each classifier in a dictionary is defined by a frontier surface in R^d. This frontier can be a line, a plane, a rectangle, etc. That means in each example the classifier is defined by a set of parameters. Thus, in order to determine the classifier $\hat{h}_H$ that minimizes the empirical error rate, given a specific dataset, we first established a grid of possible values for each one of the parameters defining the classifier, and our program made a search for the minimizing vector of parameters. Next, there was a refinement of that search in the neighbourhood of the first solution for that minimization problem. Mainly in the case of the more complex models, the runtime of these minimizing searches became very large, and this was one of our main computational challenges. Of course, the efficiency of our programs might be significantly improved if we could implement better optimizing algorithms.

III. EVALUATION OF TRADE OFF BETWEEN FEATURES AND SAMPLES: AN SLT APPROACH

A. DESCRIBING THE PROBLEM TO BE INVESTIGATED

Recall that, as stated in II-A, we have a classification problem with two classes, where:

- The random vector (X,Y) follows a joint probability distribution P.
- X is a d-dimensional vector of features.
- The response Y can be either – 1 or +1.
- We only consider classifiers in a specific dictionary H (e.g., the set of linear classifiers).

Our goal is to use a dataset D with n pairs (x_i, y_i) in order to choose a "good" classifier h in H. But what do we mean by a good classifier?

As we saw in II-A, the lower its expected error rate, the more efficient will a classifier be considered. And, since we are working with simulated data, the best way to evaluate a classifier's expected performance is calculating its theoretical loss. Now we will investigate a possible trade-off between dimensionality and cardinality in the context of a binary classification problem.

The idea is to compare two situations A and B, such that:

- In situation A, we have dimensionality = d_A and cardinality = n_A;
- In situation B, we have dimensionality = d_B and cardinality = n_B.
- All features present in A are also present in B.

If $d_A < d_B$, can the smaller dimensionality of situation A be compensated by an increase in the dataset cardinality, that is, by making $n_A > n_B$?

Of course, answering this question is not just a matter of comparing dimensionalities and cardinalities. Suppose we consider adding new features to dataset A in order to increase its discrimination capacity. Indeed, there are features that, when added, significantly improve the model's discrimination power, while there are also other features whose addition would be practically useless for that purpose. So, it is not just a matter of how many new features are added to our dataset. It depends on the effective contribution of each new feature in terms of increasing the model's discriminative power between the two populations involved in the problem. Incidentally, in all six basic examples of this paper, the discrimination power of each feature is the same. And later we will see that this fact helps making more meaningful any considerations about the number of active features in a specific context.

B. LINEAR CLASSIFIERS: HOW CARDINALITY AFFECTS ERROR RATE IF WE USE 1, 2 OR 3 FEATURES?

In order to investigate this issue, again we considered two examples from Section 2, namely, LIN2 (two-dimensional feature space) and LIN3 (three-dimensional feature space).

Both of them involve linear classifiers.
In each of these examples:

- By construction, all available features have exactly the same discriminating power.
- The data were generated by simulation, the program looks for the separating line or plane that minimizes the empirical error rate, and the average rate of classification error was estimated from 100 replications of this process.

Example LIN2:
Two situations were compared:

A) Only feature x_2 was kept present, that is, feature x_1 was discarded (recall that here the Bayes error rate is 0.308538).
B) Both features x_1 and x_2 were kept present (here the Bayes error rate is 0.23975).

For both situations, we analyzed the average classification error rate evolution, for different values of the dataset cardinality (n = 100, 300, 1000, 3000, 10000). The results can be seen in Table 3 and Figure 8. Since in our simulations dataset size n grows exponentially, its logarithm was depicted in the horizontal axis.

Table 3: Example LIN2: Average test error rate (ATER) as a function of cardinality

n	$log_{10}(n)$	ATER(2 features)	ATER(1 feature)	Std dev(ER)
100	2	0.3087	0.2857	0.05
300	2.477	0.269	0.294	0.0289
1000	3	0.241	0.303	0.0158
3000	3.477	0.239	0.306	0.00913
10000	4	0.240	0.306	0.005

Comment: The results seem to show that, as the dataset cardinality increases, this always leads to a slow average test error rate decrease. Note that in situation A (where only x_2 is present) the average test error rate decreases with n, and asymptotically approaches the Bayes error rate = 0.308538. On the other hand, in situation B (with both features x_1 and x_2 present), the average test error rate decreases with n, asymptotically approaching the Bayes error rate = 0.23975.

Example LIN3:
The idea was to compare three situations:

A) Only feature x_3 was kept present, while features x_1 and x_2 were discarded (here the Bayes error rate is 0.158655).
B) Only features x_3 and x_2 were kept present, while feature x_1 was discarded (here the Bayes error rate is 0.07865).
C) All 3 features were kept present (here the Bayes error rate is 0.041632).

Again, for all three situations, we analyzed the average classification error rate evolution, for different values of the dataset cardinality (n = 100, 300, 1000, 3000, 10000).

The results can be seen in Table 4 and Figure 8.

Table 4: Example LIN3: Average test error rate as a function of cardinality

n	$log_{10}(n)$	ATER(3 feat)	ATER(2 feat)	ATER(1 feat)
100	2	0.021	0.0552	0.1288
300	2.477	0.0304	0.0639	0.147
1000	3	0.0353	0.0729	0.155
3000	3.477	0.0382	0.0763	0.156
10000	4	0.0402	0.0773	0.158

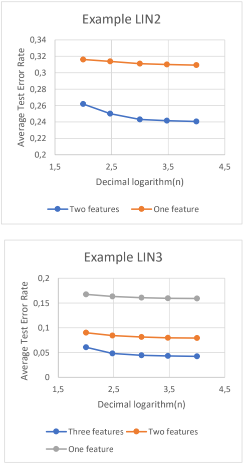


Figure 8: LIN2 and LIN3 - Average Test Error Rate versus Cardinality

Comment: In this case, again the results show there is a slow decrease in the average test error rate as the dataset cardinality grows. Note that in situation A (only x_3 present) the error rate decreases with n, asymptotically approaching the Bayes error rate = 0.1558655. In situation B (x_3 and x_2 present), it decreases with n, asymptotically approaching the Bayes error rate = 0.07865. Finally, in situation C (all features present), it also decreases with n, asymptotically approaching the Bayes error rate = 0.041632.

Summing up, these conclusions apparently show that increasing the dataset cardinality indeed has the effect of promoting an improvement, i.e., an average test error rate reduction. However, at least in our particular simulated experiment, this was a very slow reduction. Consequently, it would hardly be able to compensate the loss caused by lacking some important features.

C. EXAMPLE LIN2: ONE VERSUS TWO FEATURES. HOW CARDINALITY AFFECTS ΔL_1 AND ΔL_2?

Of course, the expected error rate is the most significant indicator of a classifier's efficiency. But another possible comparison criterion would be looking at the error rate measurement accuracy. If this were the case, then our target would be the same we were concerned with in II-B. In other words, we would prefer classifiers for which both ΔL_1 and ΔL_2 are as small as possible. Now, suppose again that, among datasets A and B, all the features in A are also present in B, i.e., $d_A \leq d_B$. Then, it would probably be easier solving the classification problem based on dataset A (lower dimension) than the one based on dataset B (higher dimension), because the complexity is bigger in high dimensional problems than it is in low dimensional problems. Consequently, we expect high dimensional problems to require larger datasets if we want to keep the same accuracy level in both cases. Therefore, we can say that, if both datasets have the same cardinality, i.e., $n_A = n_B$, and $d_A < d_B$, then:

- We would get smaller error rates using dataset B than using dataset A;
- But, on the other hand, we would get more accurate estimates of the error rate using dataset A than using dataset B, because there is more uncertainty (i.e., more noise) present in higher dimensional datasets.

In order to illustrate this phenomenon, we built an experiment considering again Example LIN2, where the 2 features are equally helpful in their discriminating power, and the dictionary is composed by linear classifiers. Recall that in this example, conditioned on label Y = -1, X follows a binormal distribution centred in (1,0) with identity covariance matrix; and, conditioned on label Y = +1, X is a binormal centred in (0,1) with identity covariance matrix. Besides that, the priors are equal for labels -1 and +1. Now:

- In Situation A, we will use only feature x_2 to build a classifier, defined by a cut point c: if $x_2 > c$, assign label +1; otherwise, assign label -1.
- In Situation B, we will use both features x_1 and x_2 to build a classifier, defined by parameters a and b: if $x_2 > a + bx_1$, assign label +1; otherwise, assign label -1.

For Situation A and for each dataset size n = 100, 200, 400, 800, 1600, 3200, 6400, 12800, we generated by simulation 100 datasets, and for each dataset we calculated ΔL_1 and ΔL_2. By doing so, we got 100 values for ΔL_1 and 100 values for ΔL_2. Based on those, we calculated estimates for four indicators:

- med (ΔL_1) = median (approximation error)
- IQR (ΔL_1) = Interquartile distance (approximation error)
- med (ΔL_2) = median (stochastic error)
- IQR (ΔL_2) = Interquartile distance (stochastic error)

The same procedure was also followed for Situation B. The detailed results are in Appendix G. Based on these results, we were able to build Table 5, which enables us to answer the following question, in the context of Example LIN2: Suppose with only 1 feature, we use a dataset with n_1 observations. Then, in order to achieve the same precision level with both features present, which minimum dataset size n_2 should we use?

Table 5: If we move from d = 1 to d = 2, how larger must be the dataset in order to keep the same accuracy level?

IND	n_1	n_2
med(ΔL_1)	400	1321
	1600	8232
	6400	$>$ 12800
IQR(ΔL_1)	400	798
	1600	3104
	6400	11327
med(ΔL_2)	400	620
	1600	2806
	6400	7503
IQR(ΔL_2)	400	485
	1600	2173
	6400	10055

Now, suppose we are facing a concrete situation, where a small number of features were used in order to build an adequate classifier. Next, we would like to study a more detailed situation involving a larger number of features, and we want to calculate how much larger the new dataset size should be for us to keep the same accuracy level as before, in terms of one of these four above indicators. An approach like the one we just presented could be helpful in providing answers to such questions.

Summing up, as dataset cardinality n increases, two simultaneous effects occur upon the classifier's error rate:

- There is a reduction of its expected value;
- And there is also an improvement in its precision measurement.

But the second effect is much stronger than the first one.

IV. THE ROLE OF THE DICTIONARY CHOICE

A. CARDINALITY VERSUS DIMENSIONALITY AND THE DICTIONARY CHOICE ADEQUACY

Besides a possible lack of available information in terms of both dimensionality and cardinality, another aspect that could damage the classifier performance is an eventual dictionary choice inadequacy. What happens when we make a wrong choice of the dictionary to be used in our classification problem? This is a mistake that can be made while facing a complex situation in a high-dimensional space, for which we have no clear idea about the spatial arrangement of the point clouds.

Let us illustrate that point with a simple 3-dimensional example. Think of a problem with d = 3, in which: the conditional distribution of X given Y = +1 is a mixture of two 3-normals; and the conditional distribution of X given Y = - 1 is also a mixture of two other 3-normals. Suppose these four 3-normals are such that:

- Any scatterplot resulting from the projection of n points onto a plane formed by two of the coordinate axes (x_1 versus x_2, x_1 versus x_3 or x_2 versus x_3) always suggests it would be inadequate to use linear classifiers.
- And yet, there is a plane (unknown for us) in R^3, the 3-dimensional space, capable of quite successfully separating the point cloud relative to Y = +1 from the point cloud relative to Y = - 1.

In such a situation we could be wrongly induced by the 2-dimensional scatterplots to choose a more complex dictionary (perhaps to use rectangular classifiers), and this inadequate choice could possibly lead us to overfitting. If we use rectangular classifiers in a problem for which the appropriate choice would be using linear classifiers, we would probably be overfitting, since the specification of a rectangular classifier requires estimating a larger number of parameters. If, even in a low dimensional space such a misleading situation could lead us to a wrong dictionary choice, let alone in a high dimensional problem.

What about the opposite mistake? If we use linear classifiers, when the problem would require using rectangular classifiers, that would be underfitting. Incidentally, note that in our RECT2 and RECT3 previous examples, clearly the use of a linear classifier would be a complete disaster.

On the other hand, how could we relate that overfitting/underfitting discussion due to a bad dictionary choice to the point we were dealing with in III, namely, the trade-off between cardinality and dimensionality?

In order to examine this topic, we will go back to one of the six simple examples we have introduced in II-D. Specifically, we developed some experiments mixing LIN (linear examples) with RECT (rectangular examples). We will generate our data according to the following theoretical model:

X is 3-dimensional;
$P(Y=1) = P(Y=-1) = \frac{1}{2}$ (equal priors)
X is $N((1,1,1); I_{3\times3})$, if $Y = +1$;
X is $N((-1,-1,-1); I_{3\times3})$, if $Y = -1$

In this context, given the specific data generating model, the most suitable dictionary choice would clearly be using linear classifiers (straight lines or planes). So, here the choice of rectangular classifiers (rectangles or parallelepipeds) is inadequate.

Four scenarios were considered, mixing two possible challenges: eventually missing a feature and eventually choosing an inadequate dictionary. They are all described in Table 6. From now on, MTL stands for "Minimal Theoretical Loss".

Table 6: Scenarios description

Scen	Features	H boundaries	Opt theor classif in H	MTL in H
A	all	Planes	$x_1 + x_2 + x_3 = 0$	0.0416
B	miss x_3	Lines	$x_1 + x_2 = 0$	0.0786
C	all	Parllelppds	$(-0.82, \infty)^3$	0.0892
D	miss x_3	Rectngls	$(-0.54, \infty)^2$	0.112

First, let us compare the four scenarios only looking at their minimum theoretical loss in the chosen H dictionary:

- If in addition to missing the x_3 feature, we choose an inadequate dictionary (Scenario D), the minimal theoretical loss turns out to be the greatest of them all: 0.112.
- If the 3 features are present, but the dictionary choice remains inadequate (Scenario C), the minimal theoretical loss drops to 0.0892.
- If we move to the correct dictionary choice, but continue to miss feature x_3 (Scenario B), the improvement is even more expressive, since the minimal theoretical loss decreases to 0.0786.
- Finally, if the dictionary choice is correct and all 3 features are present (Scenario A), we will have the lowest minimal theoretical loss: 0.0416.

For each scenario, simulations were performed using 5 different dataset cardinalities: n = 100, 300, 1000, 3000, 10000. In each simulation we generated n points in d-space and found the classifier $\hat{h}^{(D)}$ in dictionary H minimizing the empirical loss for that particular dataset. Then, we built Tables 7 and 8, showing how the average Error Rate (for 100 simulations) corresponding to classifier $\hat{h}^{(D)}$, simultaneously depends on both the scenario and the dataset cardinality n. The difference between these two tables is in their output:

- In the first one it is the training error rate, i.e., the empirical error rate;
- Whereas in the second one it is the test error rate, i.e., the theoretical error rate.

Note: If we were dealing with real data (instead of simulated data), the test error rate would usually be assessed by splitting the dataset into train and test, or by using a resampling method. However, since here we have simulated data coming from a known theoretical model, it is possible to calculate the test error rate through a purely probabilistic approach.

Table 7: Train Error Rate as function of Scenario and dataset cardinality

Cardinality n	Scenario A	Scenario B	Scenario C	Scenario D
100	0.0201	0.0562	0.0544	0.0786
300	0.03	0.0671	0.0720	0.0947
1000	0.0366	0.0735	0.0814	0.103
3000	0.0386	0.0767	0.0859	0.108
10000	0.0399	0.0781	0.0886	0.111
MTL in H	0.0416	0.0786	0.0892	0.112

Table 8: Test Error Rate as function of Scenario and dataset cardinality

Cardinality n	Scenario A	Scenario B	Scenario C	Scenario D
100	0.0574	0.0919	0.107	0.126
300	0.0493	0.0856	0.0987	0.119
1000	0.0443	0.0817	0.0932	0.115
3000	0.0428	0.0799	0.091	0.114
10000	0.0423	0.0793	0.0901	0.113
MTL in H	0.0416	0.0786	0.0892	0.112

Figures 9 and 10 show, for each of our four scenarios, the average behavior of the train (orange) and test (blue) error rates corresponding to classifier $\hat{h}^{(D)}$, as the dataset cardinality n increases.

Comments:

1) For each scenario, as n grows indefinitely, the average train error rate increases, while the average test error rate decreases, and both of them approach the minimal theoretical loss within dictionary H. Therefore, as the dataset cardinality n increases, the distance between the train and test error rates steadily decreases towards zero.
2) Of course, the most significant quality indicator in a classification problem is the test error rate, which ideally should be as low as possible. The results show that, in any of our four scenarios, in fact there is an improvement (reduction) in the average test error rate as n grows. However, the improvement significance is not enough for us to conclude that the absence of a feature could be compensated by an increase in the

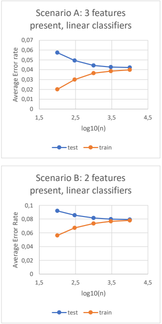


Figure 9: Scenarios A and B - Average Train and Test Error Rate versus Cardinality

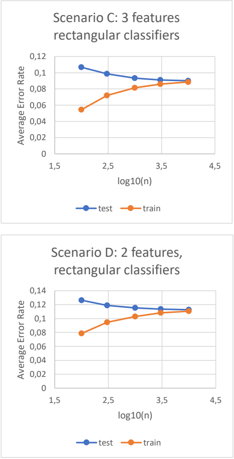


Figure 10: Scenarios C and D - Average Train and Test Error Rate versus Cardinality

dataset cardinality. On the other hand, investing in an appropriate dictionary choice could represent a truly significant gain when it comes to promoting test error rate reductions.

3) For each one of the four scenarios A, B, C and D, there is a curve of test error versus dataset size. As shown in Figure 11 a comparison of these four curves shows that vertically they are pretty separate from each other. Only for scenarios B and C, there are combinations of dataset sizes capable of producing the same test error rate level. For example, both "scenario B with cardinality = 100" and "scenario C with cardinality = 3000" lead to same average test error rate, namely, something around 0.091. This finding reveals that the choice of a wrong dictionary (as in scenario C) might cause more damage than the absence of a feature (as in scenario B), since in order to get equal error rates it would be necessary to use a dataset size 30 times larger in scenario C than in scenario B.

4) Finally, for each scenario, results were also obtained regarding the behavior of random variables ΔL_1 and ΔL_2 as a function of n (see Tables 9, 10, 11 and 12). Specifically, the median and the interquartile distance of both ΔL_1 and ΔL_2 tend to zero as n grows. These results confirm something already known about their behavior as n goes to infinity, as can be seen in Appendix A. And that prevails regardless of how many features were in fact active and how adequate the dictionary choice was for the particular classification problem we must solve.

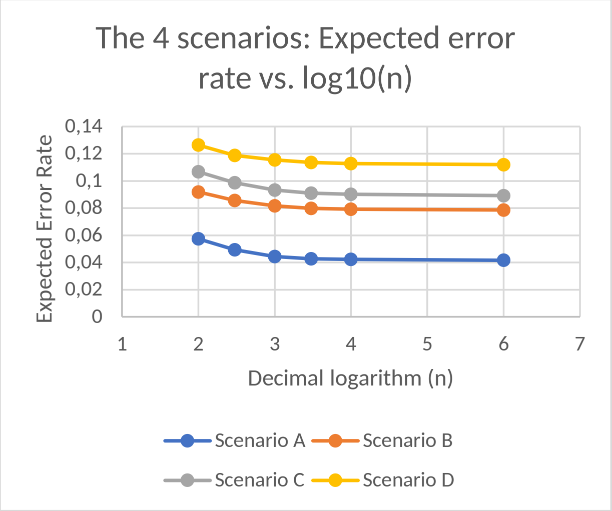


Figure 11: The four scenarios - Average Test Error Rate versus Cardinality

Table 9: Scenario A - Median and IQR for both ΔL_1 and ΔL_2 as n grows

Cardinality	med(ΔL_1)	IQR(ΔL_1)	med(ΔL_2)	IQR(ΔL_2)
100	0.01294	0.018928	0.012940	0.01998
300	0.005330	0.006900	0.005330	0.01420
1000	0.002111	0.002566	0.002111	0.00666
3000	0.001076	0.001395	0.001076	0.00408
1000	0.000458	0.000582	0.000459	0.00265

Table 10: Scenario B - Median and IQR for both ΔL_1 and ΔL_2 as n grows

Cardinality	med(ΔL_1)	IQR(ΔL_1)	med(ΔL_2)	IQR(ΔL_2)
100	0.009191	0.017819	0.03992	0.02943
300	0.003758	0.008949	0.01900	0.01945
1000	0.002072	0.002838	0.00883	0.00857
3000	0.000945	0.001661	0.00422	0.00542
10000	0.000332	0.000646	0.00179	0.00264

Table 11: Scenario C - Median and IQR for both ΔL_1 and ΔL_2 as n grows

Cardinality	med(ΔL_1)	IQR(ΔL_1)	med(ΔL_2)	IQR(ΔL_2)
100	0.012409	0.015812	0.012409	0.03048
300	0.007009	0.007518	0.007009	0.01858
1000	0.003729	0.004777	0.003729	0.00904
3000	0.001307	0.001940	0.001307	0.00462
10000	0.000638	0.000852	0.000638	0.00321

Table 12: Scenario D - Median and IQR for both ΔL_1 and ΔL_2 as n grows

Cardinality	med(ΔL_1)	IQR(ΔL_1)	med(ΔL_2)	IQR(ΔL_2)
100	0.012190	0.015590	0.05309	0.03331
300	0.006020	0.008081	0.02496	0.01982
1000	0.002010	0.003049	0.01251	0.00900
3000	0.001067	0.002297	0.00538	0.00599
10000	0.000627	0.000887	0.00217	0.00303

B. SOME GENERAL PROPERTIES OF THE EXPECTED ERROR RATES

The asymptotic behavior of the error rate, as it is described in Comment 1 of IV-A, is a consequence of some general properties, formulated and proven below:

For every dataset cardinality n:

(a) The expected value of the training (i.e., empirical) error rate is less than or equal to the minimum theoretical error rate within dictionary H. In symbols, $E(\hat{L}(\hat{h}^{(D)})) \leq \min_{h \in H} L(h)$.

(b) The expected value of the test (i.e. theoretical) error rate is greater than or equal to the minimum theoretical error rate within dictionary H. In symbols, $\min_{h \in H} L(h) \leq E(L(\hat{h}^{(D)}))$.

(c) As the dataset cardinality n tends to infinity, both expected values of the error rate, training and test, asymptotically approach the minimum theoretical error rate within dictionary H. In symbols, $\lim_{n \to \infty} E(\hat{L}(\hat{h}^{(D)})) = \lim_{n \to \infty} E(L(\hat{h}^{(D)})) = \min_{h \in H} L(h)$

Proof:

Let n be a positive integer.

(a) According to $\hat{h}^{(D)}$ definition, $\hat{L}(h) \geq \hat{L}(\hat{h}^{(D)})$, for every h ∈ H and for all dataset D whose size is n. Then, $E(\hat{L}(h)) \geq E(\hat{L}(\hat{h}^{(D)}))$, for every h ∈ H. Furthermore, $L(h) = E(\hat{L}(h))$, for all h ∈ H. Therefore, $E(\hat{L}(\hat{h}^{(D)})) \leq \min_{h \in H} L(h)$.

(b) On the other hand, $\min_{h\in H} L(h) \leq L(\hat{h}^{(D)})$, for all dataset D. Then, $\min_{h\in H} L(h) \leq E(L(\hat{h}^{(D)}))$.

(c) From now on, in all convergence statements, n goes to infinity. In Appendix A we showed that $\Delta L_1 = L(\hat{h}^{(D)}) - \min_{h\in H} L(h)$ converges to zero in probability. This implies $E(L(\hat{h}^{(D)}))$ tends to $\min_{h\in H} L(h)$. Again, from Appendix A, it is also known that $\Delta L_2 = |L(\hat{h}^{(D)}) - \hat{L}(\hat{h}^{(D)})|$ converges to zero in probability. And this enables us to prove that $\hat{L}(\hat{h}^{(D)})$ also converges to $\min_{h\in H} L(h)$ in probability. So, finally, we can conclude that $E(\hat{L}(\hat{h}^{(D)}))$ tends to $\min_{h\in H} L(h)$.

Obs.: In order to prove part (c), we twice used the following easily demonstrable property: If the sequence $X_1, X_2, \ldots$ of random variables tends in probability to the constant a, and there is a common upper bound M for all of them (i.e., $|X_n| \leq M$, for all n), then $\lim_{n\to\infty} E(X_n) = a$.

Besides that, our simulations also indicate that, as n grows towards infinity, then both non-negative differences:

$$\Delta L_{test} = E(L(\hat{h}^{(D)})) - \min_{h\in H} L(h) \text{ and}$$
$$\Delta L_{train} = \min_{h\in H} L(h) - E(\hat{L}(\hat{h}^{(D)}))$$

tend to zero as fast as $\frac{1}{\sqrt{n}}$. Unfortunately, we still don't have a formal proof for this statement.

But, in order to provide some evidence supporting this conclusion, based on the results from Tables 7 and 8 relative to our four Scenarios, in Figure ... we plotted estimates for both differences ΔL_{test} and ΔL_{train} against $\frac{1}{\sqrt{n}}$.

V. CONCLUSION

In this article we analysed the joint effect of changes in: dataset cardinality n, number of active features d, dictionary H choice, etc. on a classifier's expected error rate.

For that purpose, we adopted an experimental approach based on simulated data. Six simple examples of theoretical classification problems (See II-D) were created, discussed, analysed and simulated, using different dataset sizes. The dimensionality d of the feature space varied from 1 to 3, and the dictionary boundaries were either hyper-planes or hyper-rectangles. The relative analytical simplicity of the probabilistic models played a key role in this approach, by enabling us to easily compute almost all theoretical losses.

Indeed, our experimental approach showed that increasing dataset cardinality could compensate an eventual damage caused by lacking some important features. However, the expected error rate decreases very slowly as dataset cardinality grows. Consequently, the so-called trade-off between cardinality and dimensionality by itself seems to be, at least, controversial (See III).

Besides that, we also investigated the interaction between the cardinality/dimensionality trade-off and the dictionary choice adequacy (See IV-A). And, for that purpose, we created and simulated four different scenarios. Our simulation results seemed to show that, in order to guarantee the success of a classifier's performance, making an appropriate dictionary choice could be even more relevant than adding new features to the current dataset. Also, we proved some theoretical results about the expected error rate evolution as dataset cardinality grows (See IV-B).

Furthermore, in Appendix A we also proved that, if the VC dimension of dictionary H is proportional to the dimensionality d of the feature space, then both random variables: the stochastic error ΔL_1 and the estimation error ΔL_2 converge in probability to zero, as n tends to infinity. This is relevant because it implies that, if this assumption prevails, then as n grows towards infinity, simultaneously the stochastic error and the estimation error go to zero, and both convergence behaviours happen with a probability that grows to 1.

Our experimental results also show that, as n grows, the empirical cumulative distributions of both ΔL_1 and ΔL_2 become closer and closer to a step function at zero, for the six examples and four scenarios already mentioned. (See II-D and IV-A and also Appendix D) Clearly, this evidence confirms we do have convergence in probability to zero, for both random variables. This was indeed what we expected to see, since in all those cases our basic assumption prevails, namely, the VC dimension of H is proportional to the dimensionality of the feature space.

The article also contains a few appendices dealing with different aspects of our research, such as: how to get lower and upper limits for Bayes error rate (Appendix E), how to get tighter upper bounds for random variables ΔL_1 and ΔL_2 (Appendix F), how to use Monte Carlo methods in order to calculate the Bayes error rate in some specific cases, etc (Appendix C).

.

APPENDIX A WHY ΔL_1 AND ΔL_2 CONVERGE IN PROBABILITY TO ZERO, AS N GOES TO INFINITY?

Theorem 9.1 of Giraud's book states:
For all $t > 0$, with probability at least $1 - e^{-t}$, we have:

$$|L(\hat{h}_H) - \hat{L}_n(\hat{h}_H)| \leq 4\sqrt{\frac{2log(2S_n(H))}{n}} + \sqrt{\frac{2t}{n}} \text{ and}$$
$$L(\hat{h}_H) - \min_{h\in H} L(h) \leq 2\sqrt{\frac{2log(2S_n(H))}{n}} + \sqrt{\frac{t}{2n}}$$

where $S_n(H)$ stands for dictionary H shattering coefficient.

Notice that the classifier $\hat{h}_H$ depends on our particular dataset. So, among the three above error rates, only the first one is a fixed parameter, while the other two are random quantities. Consequently, the left-hand side of both above inequalities are random variables. Also, according to Sauer's Lemma (Proposition 9.6 in Giraud's book), $S_n(H) \leq (n+1)^{d_H}$, where d_H is the VC dimension for dictionary H. Substituting this last inequality in the two inequalities of Theorem 9.1, we can conclude that, with probability at least $1 - e^{-t}$:

$$|L(\hat{h}_H) - \hat{L}_n(\hat{h}_H)| \leq RHS(n,t) \qquad \text{and}$$
$$L(\hat{h}_H) - \min_{h \in H} L(h) \leq 2RHS(n,t), \qquad (*)$$
$$\text{where } RSH(n,t) = 2\sqrt{\frac{2ln(2) + d_H ln(n+1)}{n}} + \sqrt{\frac{t}{2n}}$$

Now, if:

- the VC dimension d_H of H is proportional to d; and
- t is proportional to log(n),

then, as $n \to \infty$, both $(1 - e^{-t} \to 0)$ and $(RHS(n,t) \to 0)$. Consequently, if d is fixed, then as $n \to \infty$, the left-hand side of both inequalities in (*) tend to zero, with a probability that tends to 1. In fact, here we are talking about convergence in probability. In order to simplify the notation, we will create two non-negative random variables:

$$\Delta L_1 = L(\hat{h}_H) - \min_{h \in H} L(h) \qquad \text{and}$$
$$\Delta L_2 = |L(\hat{h}_H) - \hat{L}_n(\hat{h}_H)|$$

We claim that both ΔL_1 and ΔL_2 converge to zero in probability, as n grows indefinitely. Let us consider first the case of ΔL_2. According to the definition of convergence in probability, that means: For every $\epsilon > 0$ and $\delta > 0$, there is an integer n_0 such that:

$$\text{if } n \geq n_0, \text{ then } P(\Delta L_2 > \epsilon) < \delta.$$

Proof: Let ϵ and δ be two positive arbitrary real numbers. We know that:

$$P(\Delta L_2 \leq RHS(n,t)) \geq 1 - e^{-t}, \text{ where}$$
$$RSH(n,t) = 2\sqrt{\frac{2log(2S_n(H))}{n}} + \sqrt{\frac{t}{2n}}$$

So, if we make $t = ln(n)$, it follows that:

$$P(\Delta L_2 \leq RHS(n,t)) \geq 1 - \tfrac{1}{n}, \quad \text{which implies}$$
$$P(\Delta L_2 > RHS(n,t)) < \tfrac{1}{n} \qquad \text{(I)}$$

On the other hand, since $\frac{ln(n)}{n}$ goes to zero as $n \to \infty$, it is easy to see that

$$\lim_{n \to \infty} RHS(n, ln(n)) = 0, \qquad \text{(II)}$$

because d_H is assumed to be proportional to d, and d is a fixed positive integer.
So, from (I), there is an integer n_{01} such that,

$$n > n_{01} \Rightarrow P(\Delta L_2 > RHS(n, ln(n)) < \delta \qquad \text{(III)}$$

Also, from (II), there is an integer n_{02} such that,

$$n > n_{02} \Rightarrow RHS(n, ln(n)) < \epsilon. \qquad \text{(IV)}$$

Take $n_0 = max\{n_{01}, n_{02}\}$. Now, let n be any integer, with $n > n_0$. Then, both (III) and (IV) are true.
So, $P(\Delta L_2 > \epsilon) < P(\Delta L_2 > RHS(n, ln(n)) < \delta$, qed.

An analogous reasoning can be used to prove that random variable $\Delta L_1 = L(\hat{h}_H) - \min_{h \in H} L(h)$ also converges to zero in probability, as n goes to infinity.

APPENDIX B DESCRIBING THE SIX EXAMPLES

1) Example LIN1

X is 1-dimensional $\qquad P(Y = 1) = P(Y = -1) = \frac{1}{2}$
X is N(1; 1), given Y = +1; $\qquad$ X is N(- 1; 1), given Y = -1.

A classifier h in dictionary H is defined by just one single cut point c: It assigns label +1, if $x > c$; and label -1, otherwise.
Then, the theoretical error rate is calculated by the expression:
$P(error) = 0.5\pi(1) + 0.5\pi(-1)$, where: $\pi(1) = \Phi(c-1)$; $\pi(-1) = 1 - \Phi(c+1)$. In this case, obviously, the best theoretical classifier has its cut point at zero (here this is also the Bayes classifier). It can be shown that its theoretical error rate is: L* = $\min_{h \in H} L(h) = 1 - \Phi(1) = 0.158655$. Here $\Phi(.)$ stands for the cumulative distribution function of the standard normal model.

2) Example RECT1

X is 1-dimensional $\qquad P(Y = 1) = P(Y = -1) = \frac{1}{2}$
X is N(0; 1), given Y = +1;
X is $0.5N(-1; 1) + 0.5N(1; 1)$, given Y = -1.

A classifier h in dictionary H is defined by an interval (a,b). That decision rule h assigns label: +1, if $x \in (a,b)$; and -1, if $x \notin (a,b)$.
Then, the theoretical error rate is calculated by the expression:
$P(error) = 0.5(1 - \pi(0)) + 0.25(\pi(1) + \pi(-1))$, where: $\pi(0) = \Phi(b) - \Phi(a)$; $\pi(1) = \Phi(b-1) - \Phi(a-1)$; $\pi(-1) = \Phi(b+1) - \Phi(a+1)$.

The classifier minimizing the theoretical error rate assigns label:
+1, if $x \in (-1.09; 1.09)$; and -1, otherwise.
Its theoretical error rate is:
L* $= \min_{h \in H} L(h) = 1 - \Phi(1.09) + 0.5(\Phi(2.09) - \Phi(-0.09)) = 0.39663$

3) Example LIN2

X is 2-dimensional) $\qquad P(Y = 1) = P(Y = -1) = \frac{1}{2}$
X is $N((1,0); I_{2\times 2})$, given Y = +1;
X is $N((0,1); I_{2\times 2})$, given Y = -1.
A classifier h in dictionary H is defined by a line $x_2 = ax_1 + b$ in R^2. It assigns label: +1, if $x_2 < ax_1 + b$; and - 1, otherwise. Then, the theoretical error rate is calculated by the expression: $P(error) = 0.5(1 - \pi(1,0)) + 0.5(1 - \pi(0,1))$, where: $\pi(1,0) = \Phi(\frac{a+b}{\sqrt{1+a^2}})$; $\pi(0,1) = \Phi(\frac{1-b}{\sqrt{1+a^2}})$.
The theoretically best classifier is obviously the line $x_1 = x_2$, (which here again agrees with Bayes classifier). It can be shown that its theoretical error rate is:
$\min_{h \in H} L(h) = 1 - \Phi(\frac{\sqrt{2}}{2}) = 0.23975$.

4) Example RECT2

X is 2-dimensional $\qquad P(Y=1)=P(Y=-1)=\frac{1}{2}$
X is $N((0,0);I)$, given Y = +1;
X is $(1/4)(N((1,0);I)+N((0,1);I)+N((-1,0);I)+N((0,-1));I)$, given Y = -1,
where I is the identity 2×2 matrix
A classifier h in dictionary H is defined by a rectangle (a,b) × (c,d) in R^2 (with its sides parallel to the coordinate axes). It assigns label: +1, if $(x_1,x_2)\in(a,b)\times(c,d)$; and -1, otherwise. Then, the theoretical error rate is calculated by the expression:
$P(error)=0.5(1-\pi(0,0))+0.125(\pi(1,0)+\pi(0,1)+\pi(-1,0)+\pi(0,-1))$, where:

$$\begin{aligned}
\pi(0,0) &= (\Phi(b)-\Phi(a))(\Phi(d)-\Phi(c));\\
\pi(1,0) &= (\Phi(b-1)-\Phi(a-1))(\Phi(d)-\Phi(c));\\
\pi(0,1) &= (\Phi(b)-\Phi(a))(\Phi(d-1)-\Phi(c-1));\\
\pi(-1,0) &= (\Phi(b+1)-\Phi(a+1))(\Phi(d)-\Phi(c));\\
\pi(0,-1) &= (\Phi(b)-\Phi(a))(\Phi(d+1)-\Phi(c+1)).
\end{aligned}$$

All these quantities correspond to the conditional probability that vector X belongs to the rectangle (a,b) × (c,d). What varies from one case to the other is just the center of the two-normal distribution. The theoretically best classifier is defined by the square (- 1.34;1.34) × (- 1.34;1.34), and its theoretical error rate is: $\min_{h\in H} L(h)=$
$=\frac{1}{2}(1-(2\Phi(1.34)-1))^2)+\frac{1}{2}(\Phi(0.34)-\Phi(-2.34))(2\Phi(1.34)-1)=0.419531$
Obs.: Bayes risk, estimated by simulation with 80000 trials, was 0.418325.

5) Example LIN3

X is 3-dimensional $\qquad P(Y=1)=P(Y=-1)=\frac{1}{2}$
X is $N((1,1,1);I)$, if $Y=+1$;
X is $N((-1,-1,-1);I)$, if $Y=-1$,
where I is the identity 3×3 matrix.
A classifier h in dictionary H is defined by a plane $x_3=\alpha x_1+\beta x_2+\gamma$ in R^3. It assigns label: +1, if $x_3>\alpha x_1+\beta x_2+\gamma$; and -1, otherwise.
Then, the theoretical error rate is calculated by the expression:
$P(error)=1-0.5(\Phi(\pi(1))+\Phi(\pi(-1)))$, where:
$\pi(1)=\frac{|\alpha+\beta-1+\gamma|}{\sqrt{\alpha^2+\beta^2+1}}$; $\qquad \pi(-1)=\frac{|-\alpha-\beta+1+\gamma|}{\sqrt{\alpha^2+\beta^2+1}}$.
Theoretically, the best classifier in H (and again this is also the Bayes classifier) is given by the plane $x_1+x_2+x_3=0$ (i.e., $\alpha=\beta=-1$, $\gamma=0$), and its theoretical error rate is: $\min_{h\in H} L(h)=1-\Phi(\sqrt{3})=0.041632$.

6) Example RECT3

X is 3-dimensional) $\qquad P(Y=1)=P(Y=-1)=\frac{1}{2}$
X is $N((0,0,0);I)$, given $Y=+1$;
X is $(1/6)(N((2,0,0);I)+N((-2,0,0);I)+N((0,2,0);I)+$ $+N((0,-2,0);I)+N((0,0,2);I)N((0,0,-2);I))$, given $Y=-1$,
where I is the identity 3×3.

$$\begin{aligned}
&X \text{ is } N((0,0,0);I) && \text{, given } Y=+1\\
&X \text{ is } (1/6)\big(N((2,0,0);I)+\\
&\quad N((-2,0,0);I)+N((0,2,0);I)+\\
&\quad N((0,-2,0);I)+N((0,0,2);I)+\\
&\quad N((0,0,-2);I)\big) && \text{, given } Y=-1
\end{aligned}$$

where I is the identity 3×3.

A classifier h in H is given by a parallelepiped $(a,b)\times(c,d)\times(e,f)$. It assigns label: +1, if $(x_1,x_2,x_3)\in(a,b)\times(c,d)\times(e,f)$; and - 1, otherwise. Then, the theoretical error rate is calculated by the expression: P(error) =
$=0.5(1-\pi_{0,0,0})+0.5(1/6)(\pi_{2,0,0}+\pi_{-2,0,0}+\pi_{0,2,0}+\pi_{0,-2,0}+\pi_{0,0,2}+\pi_{0,0,-2})$, where: $\pi_{0,0,0}=(\Phi(b)-\Phi(a))(\Phi(d)-\Phi(c))(\Phi(f)-\Phi(e))$
$\pi_{2,0,0}=(\Phi(b-2)-\Phi(a-2))(\Phi(d)-\Phi(c))(\Phi(f)-\Phi(e))$
$\pi_{0,2,0}=(\Phi(b)-\Phi(a))(\Phi(d-2)-\Phi(c-2))(\Phi(f)-\Phi(e))$
$\pi_{0,0,2}=(\Phi(b)-\Phi(a))(\Phi(d)-\Phi(c))(\Phi(f-2)-\Phi(e-2))$
$\pi_{-2,0,0}=(\Phi(b+2)-\Phi(a+2))(\Phi(d)-\Phi(c))$ $(\Phi(f)-\Phi(e))$
$\pi_{0,-2,0}=(\Phi(b)-\Phi(a))(\Phi(d+2)-\Phi(c+2))$ $(\Phi(f)-\Phi(e))$
$\pi_{0,0,-2}=(\Phi(b)-\Phi(a))(\Phi(d)-\Phi(c))$ $(\Phi(f+2)-\Phi(e+2))$

All these quantities correspond to the conditional probability that vector X belongs to the parallelepiped (a,b) × (c,d) × (e,f). What varies from one case to the other is just the center of the 3-normal conditional distribution. Here the best classifier is given by the cube (-1.73,1.73) × (-1.73,1.73) × (-1.73,1.73), and its theoretical error rate is: $\min_{h\in H} L(h)=$
$=0.5(2\Phi(1.73)-1)^3+0.5(2\Phi(1.73)-1)^2(\Phi(3.73)-\Phi(0.27))=0.2804578$.
Obs.: Bayes risk, estimated by simulation with 100000 trials, was 0.27904.

APPENDIX C MONTE CARLO CALCULATION OF BAYES RISK FOR EXAMPLES RECT2 AND RECT3

Let us consider Example RECT2. For one of the classes, we have a two-normal centered at the origin. For the other class, we have a mixture of 4 two-normals, whose centers are located at: (1,0), (0,1), (-1,0) and (0, -1), with equal weights. Here, the Bayes classifier is defined by a curve, symmetric in relation to the origin, something “in between a circle and a square”.

Its equation can be expressed as the locus of the points x in R^2 such that:

$$f_0(x)=(1/4)\sum_{i=1}^{4} f_i(x), \text{ where:}$$

$f_0(.)=$ density of a 2-normal distribution $N((0,0),I_{2\times 2})$;
$f_1(.)=$ density of a 2-normal distribution $N((1,0),I_{2\times 2})$;
$f_2(.)=$ density of a 2-normal distribution $N((0,1),I_{2\times 2})$;
$f_3(.)=$ density of a 2-normal distribution $N((-1,0),I_{2\times 2})$;

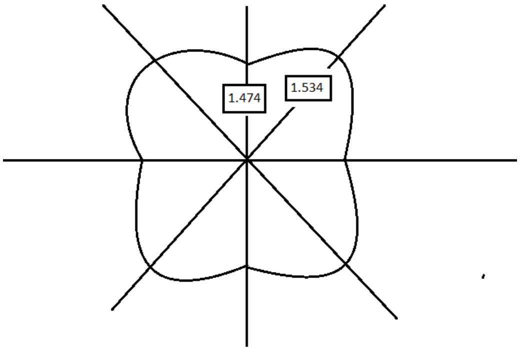


Figure 12: Bayes frontier for Example RECT2

$f_4(.)$ = density of a 2-normal distribution $N((0,-1), I_{2\times 2})$.
The distance from the origin to the point where the curve cuts each axis is 1.474.
The distance from the origin to the point where the curve cuts each line at 45^o is 1.534.
If $f_0(x) > (1/4)(f_1(x) + f_2(x) + f_3(x) + f_4(x))$, then x in inside the curve.
If $f_0(x) < (1/4)(f_1(x) + f_2(x) + f_3(x) + f_4(x))$, then x in outside the curve.

The procedure to compute the Bayes risk, works like this:

1) Set n = 80000
2) Generate, by simulation, n points $x_1, \dots, x_n$ from the distribution $D_0 = N((0,0), I_{2\times 2})$
3) Set count1 = 0
4) For each i = 1,2,. . . ,n
    - Calculate $f_0(x_i), f_1(x_i), f_2(x_i), f_3(x_i), f_4(x_i)$
    - If condition $f_0(x_i) > (1/4)\sum_{i=1}^{4} f_i(x)$ holds, count1 = count1 + 1
5) Set $\hat{P}(error|D_0) = 1 - \frac{count1}{n}$
6) Generate, by simulation, n points $x_1, \dots, x_n$ from the distribution D_m = mixture of 4 Normals.
7) Set count2 = 0
8) For each i = 1,2,. . . ,n
    - Calculate $f_0(x_i), f_1(x_i), f_2(x_i), f_3(x_i), f_4(x_i)$
    - If condition $f_0(x_i) > (1/4)\sum_{i=1}^{4} f_i(x)$ holds, count2 = count2 + 1
9) Set $\hat{P}(error|D_m)) = \frac{count2}{n}$
10) Set $\hat{P}(error) = \frac{1}{2}\hat{P}(error|D_0) + \frac{1}{2}\hat{P}(error|D_m)$

Following this procedure, we obtained $\hat{P}(error) = 0.418325$.

Now let us consider Example RECT3. For one of the classes, we have a three-normal centered at the origin. For the other class, we have a mixture of 6 three-normals, whose centers are located at: (2,0,0), (0,2,0), (0,0,2), (-2,0,0), (0,-2,0) and (0,0,-2), with equal weights. In this case, the Bayes classifier is defined by a surface in R^3, symmetric in relation to the origin, which is something “in between a sphere and a cube”. Its equation can be expressed as the locus of the points x in R^3, such that:

$f_0(x) = (1/6)\sum_{i=1}^{6} f_i(x)$, where:

- $f_0(.)$ is the density function of a three-normal distribution $N((0,0,0), I_{3\times 3})$;
- $f_1(.)$ is the density function of a three -normal distribution $N((2,0,0), I_{3\times 3})$;
- $f_2(.)$ is the density function of a three -normal distribution $N((0,2,0), I_{3\times 3})$;
- $f_3(.)$ is the density function of a three -normal distribution $N((0,0,2), I_{3\times 3})$;
- $f_4(.)$ is the density function of a three -normal distribution $N((-2,0,0), I_{3\times 3})$;
- $f_5(.)$ is the density function of a three -normal distribution $N((0,-2,0), I_{3\times 3})$;
- $f_6(.)$ is the density function of a three -normal distribution $N((0,0,-2), I_{3\times 3})$;

If $f_0(x) > (1/6)\sum_{i=1}^{6} f_i(x)$, then x in inside the surface.
If $f_0(x) < (1/6)\sum_{i=1}^{6} f_i(x)$, then x in outside the surface.

The procedure to compute the Bayes risk, works like this:

1) Set n = 100000
2) Generate, by simulation, n points $x_1, \dots, x_n$ from the distribution $D_0 = N((0,0,0), I_{3\times 3})$
3) Set count1 = 0
4) For each i = 1,2,. . . ,n
    - Calculate $f_j(x_i), j = 0, ..., 6$
    - If condition $f_0(x_i) > (1/6)\sum_{j=1}^{6} f_j(x_i)$ holds, count1 = count1 + 1
5) Set $\hat{P}(error|D_0) = 1 - \frac{count1}{n}$
6) Generate, by simulation, n points $x_1, \dots, x_n$ from the distribution D_m = mixture of 6 Normals.
7) Set count2 = 0
8) For each i = 1,2,. . . ,n
    - Calculate $f_j(x_i), j = 0, ..., 6$
    - If condition $f_0(x_i) > (1/6)\sum_{j=1}^{6} f_j(x_i)$ holds, count2 = count2 + 1
9) Set $\hat{P}(error|D_m)) = count2/n$
10) Set $\hat{P}(error) = \frac{1}{2}\hat{P}(error|D_0)) + \frac{1}{2}\hat{P}(error|D_m))$

Following this procedure, we obtained $\hat{P}(error) = 0.27904$.

APPENDIX D EXPERIMENTAL RESULTS FOR ΔL_1 AND ΔL_2 BEHAVIOUR IN THE SIX EXAMPLES

A. TABLES DESCRIBING CENTRALITY AND DISPERSION EVOLUTION OF BOTH RANDOM VARIABLES AS CARDINALITY GROWS

Now we present some tables in order to show the behaviour of random variables ΔL_1 and ΔL_2 as the dataset cardinality n grows to infinity. Here ’Med’ stands for Median and ’IQR’ stands for Interquartile Range.

Table 13: Results for Example LIN1 : ΔL_1

Dataset size	$Med(\Delta L_1)$	$IQR(\Delta L_1)$	RHS	prob
n = 100 (red)	0.004574	0.010419	0.752	0.865
n = 1000 (green)	0.000876	0.003282	0.272	0.950
n = 10000 (blue)	0.000302	0.000725	0.096	0.982

Table 14: Results for Example LIN1 : ΔL_2

Dataset size	$Med(\Delta L_2)$	$IQR(\Delta L_2)$	RHS	prob
n = 100 (red)	0.032554	0.031930	0.752	0.865
n = 1000 (green)	0.010307	0.013794	0.272	0.950
n = 10000 (blue)	0.002879	0.002966	0.096	0.982

1) Example LIN1 – One feature - Linear classifiers

2) Example RECT1 - One feature – Rectangular classifiers

Table 15: Results for Example RECT1: ΔL_1

Dataset size	$Med(\Delta L_1)$	$IQR(\Delta L_1)$	RHS	prob
n = 100 (red)	0.017762	0.019027	0.752	0.865
n = 1000 (green)	0.004087	0.004892	0.272	0.950
n = 10000 (blue)	0.000974	0.001069	0.096	0.982

Table 16: Results for Example RECT1: ΔL_2

Dataset size	$Med(\Delta L_2)$	$IQR(\Delta L_2)$	RHS	prob
n = 100 (red)	0.079528	0.056357	0.752	0.865
n = 1000 (green)	0.016137	0.017578	0.272	0.950
n = 10000 (blue)	0.003754	0.005863	0.096	0.982

3) Example LIN2 – Two features - Linear classifiers

Table 17: Results for Example LIN2: ΔL_1

Dataset size	$Med(\Delta L_1)$	$IQR(\Delta L_1)$	RHS	prob
n = 100 (red)	0.004574	0.010419	0.881	0.865
n = 1000 (green)	0.000876	0.003282	0.323	0.950
n = 10000 (blue)	0.000302	0.000725	0.115	0.982

Table 18: Results for Example LIN2: ΔL_2

Dataset size	$Med(\Delta L_2)$	$IQR(\Delta L_2)$	RHS	prob
n = 100 (red)	0.032554	0.031930	0.881	0.865
n = 1000 (green)	0.010307	0.013794	0.323	0.950
n = 10000 (blue)	0.002879	0.002966	0.115	0.982

4) Example RECT2 – Two features - Rectangular classifiers

Table 19: Results for Example RECT2: ΔL_1

Dataset size	$Med(\Delta L_1)$	$IQR(\Delta L_1)$	RHS	prob
n = 100 (red)	0.024565	0.017722	0.991	0.865
n = 1000 (green)	0.009016	0.009152	0.367	0.950
n = 10000 (blue)	0.002073	0.002237	0.131	0.982

Table 20: Results for Example RECT2: ΔL_2

Dataset size	$Med(\Delta L_2)$	$IQR(\Delta L_2)$	RHS	prob
n = 100 (red)	0.125330	0.040044	0.991	0.865
n = 1000 (green)	0.037320	0.018232	0.367	0.950
n = 10000 (blue)	0.007799	0.007017	0.131	0.982

5) Example LIN3 – Three features - Linear classifiers

Table 21: Results for Example LIN3: ΔL_1

Dataset size	$Med(\Delta L_1)$	$IQR(\Delta L_1)$	RHS	prob
n = 100 (red)	0.014100	0.015163	0.991	0.865
n = 1000 (green)	0.002444	0.002719	0.367	0.950
n = 10000 (blue)	0.000518	0.000597	0.131	0.982

Table 22: Results for Example LIN3: ΔL_2

Dataset size	$Med(\Delta L_2)$	$IQR(\Delta L_2)$	RHS	prob
n = 100 (red)	0.037306	0.020655	0.991	0.865
n = 1000 (green)	0.009605	0.008167	0.367	0.950
n = 10000 (blue)	0.001637	0.001764	0.131	0.982

6) Example RECT3 – Three - Rectangular classifiers

Table 23: Results for Example RECT3: ΔL_1

Dataset size	$Med(\Delta L_1)$	$IQR(\Delta L_1)$	RHS	prob
n = 100 (red)			1.178	0.865
n = 1000 (green)			0.440	0.950
n = 10000 (blue)			0.158	0.982

Table 24: Results for Example RECT3: ΔL_2

Dataset size	$Med(\Delta L_2)$	$IQR(\Delta L_2)$	RHS	prob
n = 100 (red)			1.178	0.865
n = 1000 (green)			0.440	0.950
n = 10000 (blue)			0.158	0.982

B. THE EMPIRICAL CUMULATIVE DISTRIBUTION FUNCTIONS (ECDF'S) OF ΔL_1 AND ΔL_2 FOR DIFFERENT DATASET CARDINALITIES

Here the goal is, for each one of our six Examples, to show the ecdf's of both ΔL_1 and ΔL_2 for n = 100, n = 1000 and n = 10000.

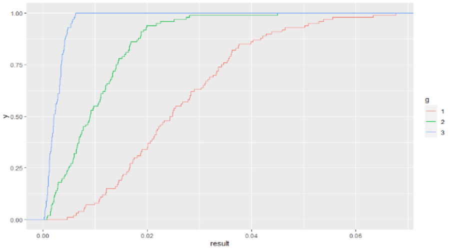


Figure 13: ecdf(ΔL_1) - LIN1

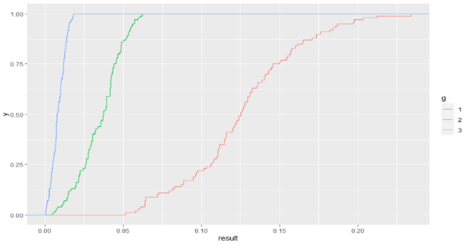


Figure 14: ecdf(ΔL_2) - LIN1

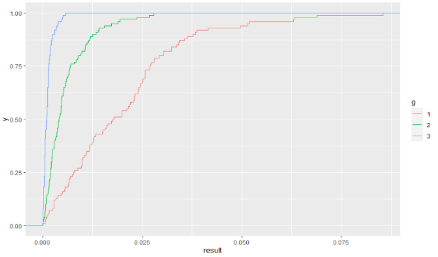


Figure 15: ecdf(ΔL_1) - RECT1

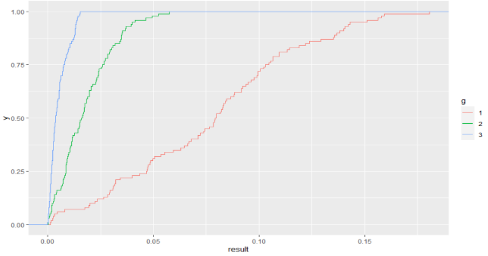


Figure 16: ecdf(ΔL_2) - RECT1

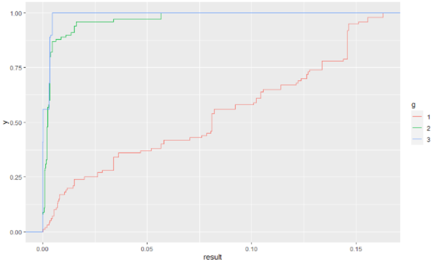


Figure 17: ecdf(ΔL_1) - LIN2

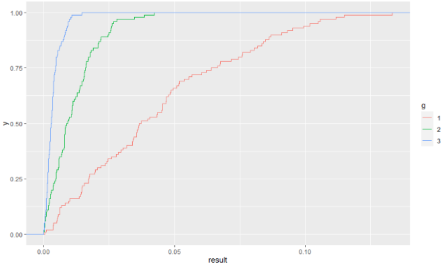


Figure 18: ecdf(ΔL_2) - LIN2

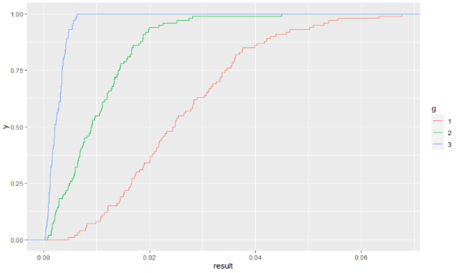


Figure 19: ecdf(ΔL_1) - RECT2

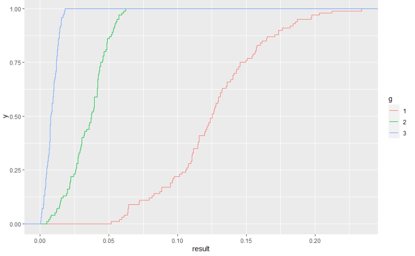


Figure 20: ecdf(ΔL_2) - RECT2

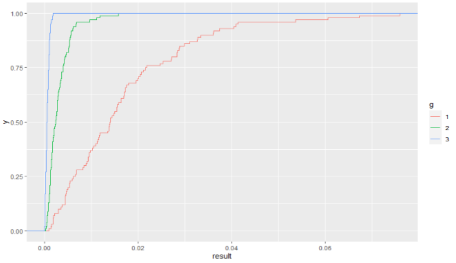


Figure 21: ecdf(ΔL_1) - LIN3

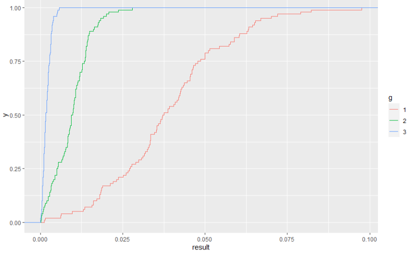


Figure 22: ecdf(ΔL_2) - LIN3

APPENDIX E LOWER AND UPPER BOUNDS FOR BAYES ERROR RATE

The article “Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure”, by Visar Berisha, Alan Wisler, Alfred O. Hero, and Andreas Spanias, presents a methodology that can be used to establish lower and upper bounds for the Bayes error rate (BER) in a binary classification problem. Clearly, this has a lot to do with our concerns in the previous sections. Therefore, in this section we report the results we got by applying this procedure to the six examples described in Section 3. Berisha et al claim such bounds can be calculated as:

$$\frac{1}{2}-\frac{1}{2}\sqrt{\tilde{D}_p(f_{-1},f_{+1})}\leq \epsilon^{Bayes}\leq \frac{1}{2}-\frac{1}{2}\tilde{D}_p(f_{-1},f_{+1})$$

The symbol $\tilde{D}_p(f_{-1},f_{+1})$ in this expression stands for a nonparametric divergence measure for the pair of conditional multivariate densities f_{-1} and f_{+1}, and it is defined as:

$$1-4p(1-p)\int_{R^d}\frac{f_{-1}(x)f_{+1}(x))}{pf_{-1}(x)+(1-p)f_{+1}(x))}dx$$

where p refers to the prior probability assigned to the +1 label. In that article, the authors also propose a method for estimating the bounds, by building the Euclidean minimal spanning tree (MST) corresponding to a full dataset generated according to the classification problem structure. The procedure was applied to each one of the six examples we presented in Section 3, namely: LIN1, RECT1, LIN2, RECT2, LIN3 and RECT3. Recall that in all of them we have p = ½. Since for both examples LIN1 and RECT1 the feature space is one-dimensional (only one feature), due to the probabilistic models simplicity, the bounds were calculated by Monte Carlo numerical integration, and the results were as follows:

Table 25: One-dimension Examples: LIN1 and RECT1

Example	BER	lower bound	upper bound
LIN1	0.158655	0.1336627	0.231594
RECT1	0.39663	0.328128	0.44092

As for the other four examples: LIN2, RECT2, LIN3 and RECT3, we used the following estimation procedure: For each one of three cardinalities: n = 100, n = 1000, n = 10000, the bounds were estimated by simulating 100 different datasets. For each generated dataset, we built the corresponding Euclidean MST, and we counted the number of tree edges joining a node with a +1 label to a node with a –1 label. This statistic, denoted by the symbol $C(X_{f_{-1}}, X_{f_{+1}})$, is the basis for calculating the bounds. Here we considered the arithmetic mean of the 100 replicate estimates. The authors prove that the random variable $1 - 2\frac{C(X_{f_{-1}}, X_{f_{+1}})}{N_{f_{-1}}+N_{f_{+1}}}$ converges to the divergence measure $\tilde{D}_p(f_{-1}, f_{+1})$, when $N_{f_{-1}}$ and $N_{f_{+1}}$, the cardinalities of the datasets, both tend to infinity, in such a way that $\frac{N_{f_{+1}}}{N_{f_{-1}}+N_{f_{+1}}}$ approaches p. The calculations were made with the Rstudio software. Here are the results we got:

Table 26: LIN2 (BER = 0.23975)

n	lower bound	upper bound
100	0.2127719	0.335
1000	0.2064698	0.32768
10000	0.204591	0.325467

Table 27: RECT2 (BER = 0.418325, $\min_{h \in H} L(h)$ = 0.419531)

n	lower bound	upper bound
100	0.4231885	0.4882
1000	0.4061917	0.4824
10000	0.3999125	0.479965

Table 28: LIN3 (BER = 0.041632)

n	lower bound	upper bound
100	0.03604957	0.0695
1000	0.03278485	0.06342
10000	0.03198184	0.061918

Table 29: RECT3 (BER = 0.27904, $\min_{h \in H} L(h)$ = 0.2804578)

n	lower bound	upper bound
100	0.2741682	0.0.398
1000	0.2514643	0.37646
10000	0.2470969	0.37208

Comments:

- Indeed, the Bayes error rate (BER) is always between the lower and the upper bounds.
- According to Berisha et al, especially with equal priors (i.e., p = q = ½), these bounds were expected to be tight. As we can see, this was not really the case here.

APPENDIX F TRYING TO GET TIGHTER UPPER BOUNDS FOR ΔL_1 AND ΔL_2

Recall that, in Section 1 we have proved the convergence in probability to zero of both random variables ΔL_1 and ΔL_2. And, for this purpose, we used Theorem 9.1 of Giraud's book, combined with Sauer's Lemma, which led us to conclude that, with probability at least $1 - e^{-t}$:

$$\Delta L_1 \leq 2RHS(n,t) \quad \text{and} \quad \Delta L_2 \leq RHS(n,t),$$

$$\text{where } RSH(n,t) = 2\sqrt{\frac{2ln(2)+d_H ln(n+1)}{n}} + \sqrt{\frac{t}{2n}}$$

These inequalities enabled us to show that, as $n \to \infty$, then both $RHS(n,t) \to 0$ and $1 - e^{-t} \to 1$. And, for this purpose, we chose t to be an increasing function of n, which grows towards infinity more slowly than n. By setting t = log(n), we were able to complete the proof. So, this reasoning really enabled us to prove the convergence in probability to zero of both ΔL_1 and ΔL_2. But, as pointed out previously, these upper bounds for ΔL_1 and ΔL_2 showed to be pretty loose. Therefore, it would be nice to obtain tighter upper bounds for both ΔL_1 and ΔL_2. Now, in the paper by Mello (Reference D) there is an alternative formulation for the statement in Theorem 9.1 of Giraud's book. Using the notation that we have been adopting in this article, Mello's approach could be formulated like this:

$$P(sup_{h\in H} \Delta L_2 > \epsilon) \leq 2S_{2n}(H)exp(-\tfrac{n\epsilon^2}{4}), \quad \text{where:}$$

$0 < \epsilon < 1$ and $S_{2n}(H)$ is the Shattering coefficient. Besides that, in the same paper, Mello also presents an inequality, by Sauer and Shelah, relating the Shattering coefficient to the VC dimension d_H of dictionary H, namely: $S_{2n}(H) \leq \sum_{i=0}^{d_H} \binom{n}{i}$. Combining these two statements, we can conclude that:

$$P(sup_{h\in H} \Delta L_2 > \epsilon) \leq 2(\textstyle\sum_{i=0}^{d_H} \binom{n}{i})exp(-\tfrac{n\epsilon^2}{4}).$$

So, setting $\alpha = 2(\sum_{i=0}^{d_H} \binom{n}{i})exp(-\frac{n\epsilon^2}{4})$, after some algebraic steps, we obtain:

$$P\left[\hat{\Delta}_n(H) \leq \sqrt{\frac{ln(2)+\sum_{i=0}^{d_H}\binom{n}{i}) - ln(\alpha)}{n}}\right] > 1 - \alpha$$

where $\hat{\Delta}_n(H) = sup_{h\in H}|L(\hat{h}_H) - \hat{L}_n(\hat{h}_H)|$.

On the other hand, according to Lemma 9.2 of Giraud´s book:

$$\Delta L_1 \leq 2\hat{\Delta}_n(H) \text{ and } \Delta L_2 \leq \hat{\Delta}_n(H)$$

Then, summing up all these statements, we can propose new upper bounds for ΔL_1 and ΔL_2: With probability $> 1-\alpha$,

$$\Delta L_1 \leq 2RHS'(n,\alpha) \quad \text{and} \quad \Delta L_2 \leq RHS'(n,\alpha),$$

$$\text{where } RHS'(n,\alpha) = 2\sqrt{\frac{ln(2)+\sum_{i=0}^{d_H}\binom{n}{i})-ln(\alpha)}{n}}$$

Now, let us see how these new upper bounds behave, as $n \to \infty$. The following table enables us to compare the old RHS and the new RHS' upper bounds in each of our six Examples, keeping the same probability levels $1-\alpha$ we had before:

Table 30: New upper bounds for ΔL_1

Example	Cardinality	Med(ΔL_1)	IQR(ΔL_1)	prob	RHS	RHS´
LIN1	100	0.004574	0.01042	0.865	0.752	0.670
	1000	0.000876	0.00328	0.950	0.272	0.259
	10000	0.000302	0.00073	0.982	0.096	0.095
RECT1	100	0.017762	0.01903	0.865	0.752	0.670
	1000	0.004087	0.00489	0.950	0.272	0.259
	10000	0.000974	0.00107	0.982	0.096	0.095
LIN2	100	0.004574	0.01042	0.865	0.881	0.767
	1000	0.000876	0.00328	0.950	0.323	0.301
	10000	0.000302	0.00073	0.982	0.115	0.111
RECT2	100	0.024565	0.01772	0.865	0.991	0.847
	1000	0.009016	0.00915	0.950	0.367	0.335
	10000	0.002073	0.00224	0.982	0.131	0.124
LIN3	100	0.014100	0.01516	0.865	0.991	0.847
	1000	0.002444	0.00272	0.950	0.367	0.335
	10000	0.000518	0.00060	0.982	0.131	0.124
RECT3	100			0.865	1.178	0.973
	1000			0.950	0.440	0.393
	10000			0.982	0.158	0.146

Table 31: New upper bounds for ΔL_2

Example	Cardinality	Med(ΔL_2)	IQR(ΔL_2)	prob	RHS	RHS´
LIN1	100	0.032554	0.03193	0.865	0.752	0.670
	1000	0.010307	0.01379	0.950	0.272	0.259
	10000	0.002879	0.00297	0.982	0.096	0.095
RECT1	100	0.079528	0.05636	0.865	0.752	0.670
	1000	0.016137	0.01758	0.950	0.272	0.259
	10000	0.003754	0.00586	0.982	0.096	0.095
LIN2	100	0.032554	0.03193	0.865	0.881	0.767
	1000	0.010307	0.01379	0.950	0.323	0.301
	10000	0.002879	0.00297	0.982	0.115	0.111
RECT2	100	0.125330	0.04004	0.865	0.991	0.847
	1000	0.037320	0.01823	0.950	0.367	0.335
	10000	0.007799	0.00702	0.982	0.131	0.124
LIN3	100	0.037306	0.02066	0.865	0.991	0.847
	1000	0.009605	0.00817	0.950	0.367	0.335
	10000	0.001637	0.00176	0.982	0.131	0.124
RECT3	100			0.865	1.178	0.973
	1000			0.950	0.440	0.393
	10000			0.982	0.158	0.146

Obs.:

- The (new and old) upper bounds for ΔL_1 are just twice larger than the ones for ΔL_2.
- The comparison between the two last columns of the table shows that the new upper bounds are indeed a little tighter than the old ones, although, as compared to the medians and IQR's of both ΔL_1 and ΔL_2, they remain pretty loose.

APPENDIX G LIN2 THE EFFECT OF A MISSING FEATURE OVER ERROR RATE PRECISION MEASUREMENT

Tables... and Figures... present, for Example LIN2, the effect over indicators $med(\Delta L1)$, $IQR(\Delta L1)$, $med(\Delta L2andIQR(\Delta L2)$, of missing or not feature x1.

Table 32: Situation A (1 feature)

n	med(ΔL_1)	IQR(ΔL_1)	med(ΔL_2)	IQR(ΔL_2)
100	0.00521	0.0108	0.0475	0.0425
200	0.00776	0.0128	0.0344	0.0304
400	0.00254	0.00602	0.0186	0.0221
800	0.00197	0.00417	0.0136	0.0131
1600	0.000878	0.00217	0.00814	0.0114
3200	0.000712	0.00269	0.00664	0.00682
6400	0.000141	0.00103	0.00441	0.00452
12800	0.000317	0.000632	0.00310	0.00349

Table 33: Situation B (2 features)

n	med(ΔL_1)	IQR(ΔL_1)	med(ΔL_2)	IQR(ΔL_2)
100	0.0120	0.0155	0.0513	0.0461
200	0.00972	0.0141	0.0464	0.0380
400	0.00763	0.00960	0.0250	0.0235
800	0.00318	0.00600	0.0133	0.0170
1600	0.00219	0.00252	0.0114	0.0133
3200	0.00174	0.00280	0.00708	0.00788
6400	0.00100	0.00141	0.00471	0.00571
12800	0.000562	0.000911	0.00300	0.00362

Using these results, we built figures 23, where:

- The blue curve corresponds to using 2 features.
- The orange curve corresponds to using only 1 feature.

Of course, if we could deal with populational quantities, those curves should always present a descending behavior as n grows. But, since we are using estimates for these four location/dispersion indicators (med(ΔL_1), IQR(ΔL_1), med(ΔL_2), IQR(ΔL_2)), it is not surprising that sometimes the above curves "oscillate" due to noise presence, mainly whenever n is small.

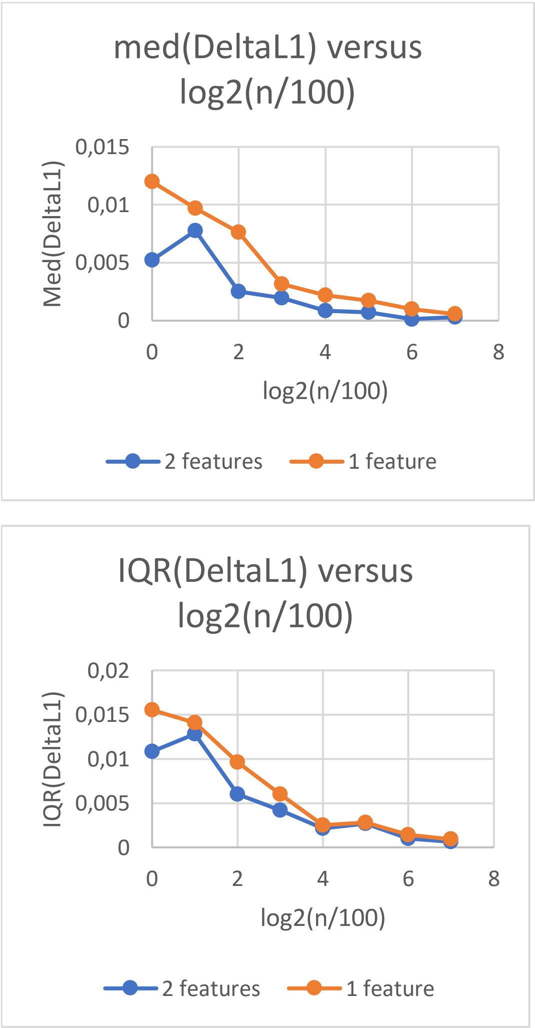
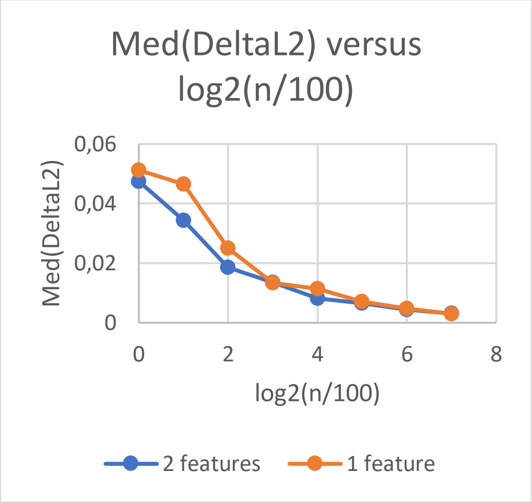
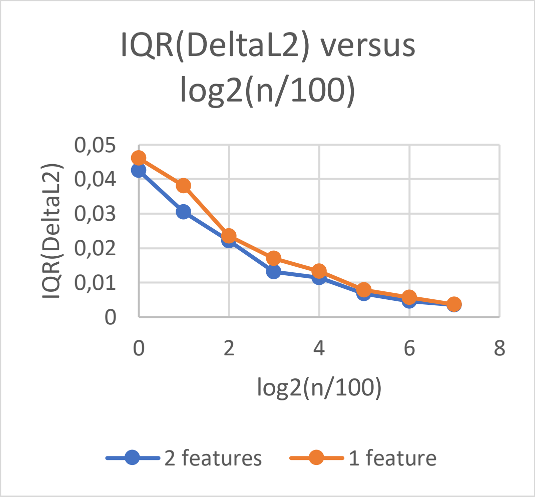


Figure 23: ΔL_1 Median and IQR versus cardinality





References

[1] C. Giraud, *Introduction to High-Dimensional Statistics*, ser. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2014. [Online]. Available: https://books.google.com.br/books?id=qRuVoAEACAAJ

[2] R. F. de Mello, C. Manapragada, and A. Bifet, "Measuring the shattering coefficient of decision tree models," *Expert Systems with Applications*, vol. 137, pp. 443–452, 2019.

[3] R. F. de Mello and M. A. Ponti, *Machine Learning - A Practical Approach on the Statistical Learning Theory*. Springer, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-94989-5

[4] R. F. de Mello, "On the shattering coefficient of supervised learning algorithms," 2020.

[5] V. Berisha, A. Wisler, A. O. Hero, and A. Spanias, "Empirically estimable classification bounds based on a nonparametric divergence measure," *IEEE Transactions on Signal Processing*, vol. 64, no. 3, pp. 580–591, 2016.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. [Online]. Available: https://faculty.marshall.usc.edu/gareth-james/ISL/

[8] J. Friedman, T. Hastie, R. Tibshirani *et al.*, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.

[9] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook New York, NY, USA:, 2012, vol. 4.

• • •