UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PAULO RENATO CARVALHO DE AZEVEDO FILHO

SLACGS: Simulator for Loss Analysis of Classifiers using Gaussian Samples

RIO DE JANEIRO
2024

PAULO RENATO CARVALHO DE AZEVEDO FILHO

SLACGS: Simulator for Loss Analysis of Classifiers using Gaussian Samples

Undergraduate thesis presented to the Institute of Computing at the Federal University of Rio de Janeiro as part of the requirements for obtaining the degree of Bachelor of Science in Computer Science.

Orientador: Prof. Daniel Sadoc Menasche

RIO DE JANEIRO
2024

PAULO RENATO CARVALHO DE AZEVEDO FILHO

SLACGS: Simulator for Loss Analysis of Classifiers using Gaussian Samples

Undergraduate thesis presented to the Institute of Computing at the Federal University of Rio de Janeiro as part of the requirements for obtaining the degree of Bachelor of Science in Computer Science.

Aprovado em 19 de março de 2024

BANCA EXAMINADORA:

Prof. Daniel Sadoc Menasché
Orientador
Ph.D. (UFRJ)

Prof. João Ismael D. Pinheiro
Co-orientador
D.Sc. (UFRJ)

Prof. João Antonio Recio da Paixão
D.Sc. (UFRJ)

Prof. Pedro Henrique Cruz Caminha
D.Sc. (UFRJ)

ABSTRACT

In this work, we developed a simulator to explore the impact of different factors on the expected error rate in binary classification problems. The simulator allows us to analyze how variations in dataset cardinality and the number of features affect the effectiveness of a classifier. Using simulated data, generated from multivariate Gaussians, we investigated the behavior of the expected error rate as a function of dataset cardinality and the number of features.

The results obtained with the simulator reveal conditions under which an increase in dataset cardinality can help compensate for the absence of features. Our results suggest that the relationship between dataset cardinality and the number of features is complex and may not be linear, highlighting a possible trade-off between the two. The simulator developed in this study thus emerges as a crucial tool for unraveling this and other dynamics involved in binary data classification.

keywords: Classifiers; SVM; Bayes Error; Performance.

LIST OF ILLUSTRATIONS

CONTENTS

1 INTRODUCTION AND MOTIVATION

In this chapter, we introduce the sample classification problem and motivate the need for a simulator to study classifier behavior. We describe key design choices in this work—including a focus on linear classifiers and Gaussian samples—and summarize our main contributions.

1.1 MOTIVATION: THE GENERAL CLASSIFICATION PROBLEM

Consider a population that can be partitioned into groups, with each group corresponding to a label. Suppose further that each element has measurable attributes and that there is an (unknown) relationship between attributes and labels.

The goal of **classification learning** is to learn a rule that assigns a label to an unlabeled observation based on its attributes. The rule is trained on a set of labeled examples (observations whose attributes and labels are known). In this context, two key factors that strongly affect performance are:

- **sample size (cardinality):** the number of observations used to train the classifier.
- **number of features (dimensionality):** the number of attributes in each observation.

Although these factors have a fundamental impact on classifier performance, their joint effect is still not fully characterized quantitatively.

One of our main goals is to build a simulator that helps quantify how factors such as sample size and dimensionality affect classification performance.

1.2 BRIEF REVIEW OF RELATED WORK

There is extensive literature on how the number of observations and the number of attributes affect classifier performance (ENTEZARI-MALEKI; REZAEI; MINAEI-BIDGOLI, 2009). However, many studies focus on very large datasets. Here we focus on small datasets—on the order of thousands of samples and one to three attributes—while noting that the simulator is generic and can also be applied to larger datasets given sufficient computational resources.

In this work, we aim to understand the joint role of the number of observations and the number of attributes in classification tasks. To this end, we built a simulator and considered one of the simplest possible scenarios, namely, focusing on the classification of observations derived from multivariate Gaussian models.

1.3 GOALS AND OVERVIEW

Our goal is to quantify how key factors influence the performance of a binary classifier.[1] To this end, we developed a flexible and extensible simulator.

The simulator allows us to analyze how different inputs affect classification quality. The main inputs we vary are:

- class means, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, for classes 1 and 2 ($\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$),
- total sample size, n, with $n/2$ observations per class,
- number of attributes (features) in each observation, D, and
- covariance structure: standard deviations σ_i and correlations $\rho_{i,j}$.

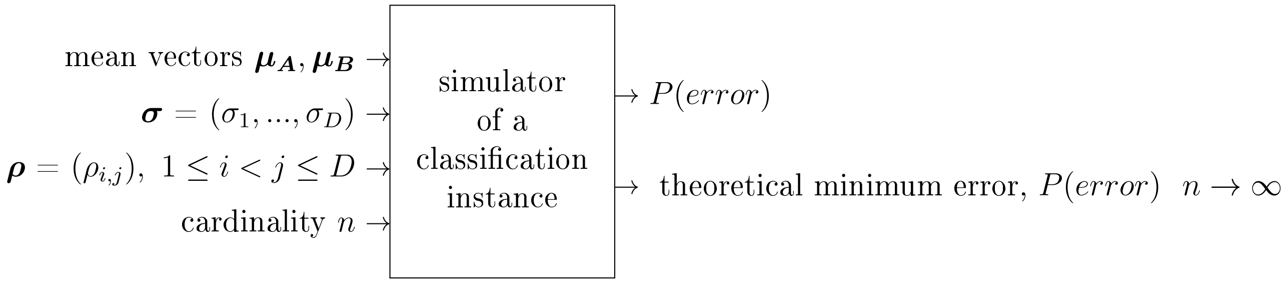

mean vectors $\boldsymbol{\mu_A}, \boldsymbol{\mu_B}$
$\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_D)$
$\boldsymbol{\rho} = (\rho_{i,j}),\ 1 \leq i < j \leq D$
cardinality n
simulator of a classification instance
$P(error)$
theoretical minimum error, $P(error)$ $n \to \infty$

Figure 1 – General scheme of the simulator for estimating the error associated with the classification of two datasets collected from Gaussian samples. Each dataset is associated with one of two classes, and each class is associated with a mean from the mean vector. We assume that the covariance matrix, constructed based on $\boldsymbol{\sigma}$ and $\boldsymbol{\rho}$, is shared between the two classes, and we consider $n/2$ observations in each class.

Figure 1 illustrates the general scheme of the simulator for a classification instance, taking into account the elements mentioned above.

1.4 RESEARCH QUESTIONS

To illustrate the results that can be obtained with the simulator, we focused on the following questions:

- When is it possible to trade off features and sample size? In other words, when additional features are unavailable, under what conditions can comparable performance be achieved by collecting more observations?
- When does adding a feature meaningfully improve classification performance?

[1] Such classifiers are also called discriminators, as they consider only two classes.

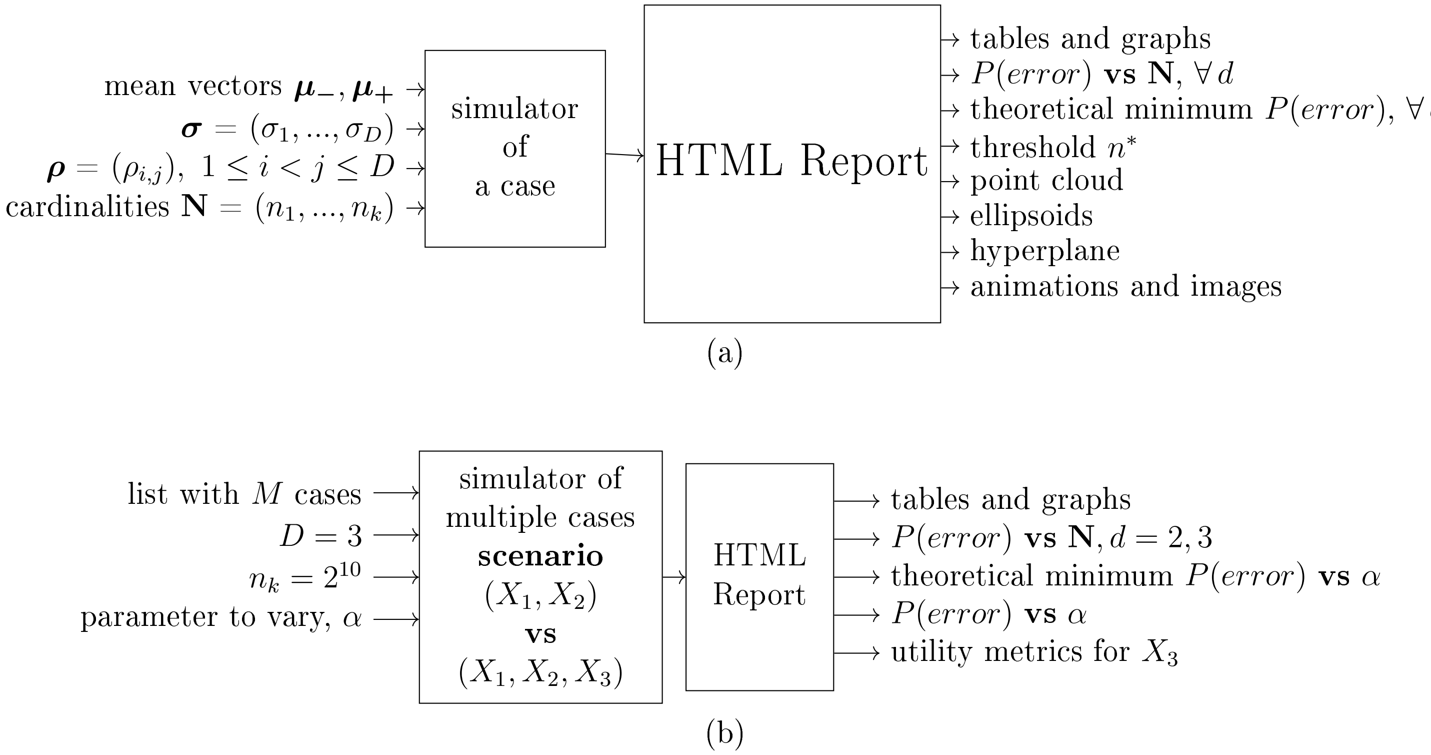

mean vectors $\boldsymbol{\mu}_-, \boldsymbol{\mu}_+$
$\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_D)$
$\boldsymbol{\rho} = (\rho_{i,j}),\ 1 \leq i < j \leq D$
cardinalities $\mathbf{N} = (n_1, ..., n_k)$
simulator of a case
HTML Report
tables and graphs
$P(error)$ **vs** $\mathbf{N}$, $\forall\, d$
theoretical minimum $P(error)$, $\forall$
threshold n^*
point cloud
ellipsoids
hyperplane
animations and images
(a)
list with M cases
$D = 3$
$n_k = 2^{10}$
parameter to vary, α
simulator of multiple cases
scenario
(X_1, X_2)
vs
(X_1, X_2, X_3)
HTML Report
tables and graphs
$P(error)$ **vs** $\mathbf{N}, d = 2, 3$
theoretical minimum $P(error)$ **vs** α
$P(error)$ **vs** α
utility metrics for X_3
(b)

Figure 2 – General scheme of the simulator: (a) case of interest, involving multiple classification instances, and (b) scenario involving multiple cases of interest, varying parameter $\alpha \in \{\sigma_3, \rho_{12}, \rho_{13}\}$, considering particularly 4 scenarios: scenario 1) varies σ_3, scenario 2) varies ρ_{12}, scenario 3) varies ρ_{13} with $\rho_{13} = \rho_{23}$, $\rho_{12} = 0$, and scenario 4) same as scenario 3 but with $\rho_{12} \neq 0$.

- What changes when moving from 1 to 2 features, or from 2 to 3 features, and when does an added feature provide a substantial gain in expected accuracy?

We implemented a simulator to generate Gaussian samples and used a Support Vector Machine (**SVM**) to classify them. We provide partial answers to the questions above. In particular, we identify scenarios where, with a small number of observations, it is advantageous to use only 2 features (rather than 3) to reduce *overfitting*. We also find a threshold number of observations such that, above $n^\star$, using all 3 features is typically advantageous.

1.5 CASES OF INTEREST AND SCENARIOS

Below, we present the cases of interest addressed by the simulator, as well as the scenarios considered in this thesis.

1.5.1 What is the impact of the number of observations and the number of features on classification error?

Next, we consider the impact of sample size and the number of features on classification error. We call this a *case of interest*.

The general scheme for analyzing a case of interest is illustrated in Figure 2(a). In this setting, we vary the number of observations over the set $\mathbf{N}$ and the number of features from 1 up to D, and we examine how both affect classification error. The output includes curves of error versus sample size, with one curve per feature count.

The resulting report allows us, for example, to assess the marginal impact of each additional feature.

In the top-right corner of Figure 2(a), we indicate typical outputs for a case of interest. These include tables and graphs such as:

- $P(error)$ as a function of the number of observations $\mathbf{N}$, for $d = 1, 2, \ldots, D$: we consider three estimators of $P(error)$ (Appendix C): theoretical error, empirical error on the training set, and empirical error on the test set;

- Theoretical minimum $P(error)$, for $d = 1, 2, \ldots, D$: this error is also known as Bayes Risk, see Appendix C;

- Threshold $n^{\star}$: the number of observations beyond which including feature X_3 becomes advantageous, assuming X_1 and X_2 are already present. See Section 3.1. Note that this threshold depends on multiple factors, including feature variances and correlations.

1.5.2 What is the impact of feature correlation and variance on classification error?

Next, we consider the impact of feature variance and correlation on classification error. We refer to this analysis as the analysis of a scenario, which consists of a set of cases of interest.

In Figure 2(b), we analyze a scenario composed of a set of cases that aim to compare various classifiers, taking into account the impact of feature variance and correlation on classification error. We can still compare a classifier with 2 features (X_1, X_2) against a classifier with 3 features (X_1, X_2, X_3). The report obtained from executing a scenario allows us to understand the influence of varying a particular parameter of the Gaussian distribution on the discriminative power of a classifier.

We observed changes in the discriminative power by adding feature X_3 in all scenarios. For this reason, we denote features X_1 and X_2 as present features, and feature X_3 as a candidate feature, as it is a candidate to be added to the set of present features.

The scenarios considered in this thesis are:

Scenario 1: We vary the dispersion parameter σ_3, that is, the variance, of feature X_3.

Scenario 2: We vary the correlation parameter ρ_{12} between features X_1 and X_2.

Scenario 3: We vary the correlation parameters ρ_{13} and ρ_{23}, which refer to the correlation between feature X_3 and feature X_1, and between feature X_3 and feature X_2. In this scenario, $\rho_{13} = \rho_{23}$ and $\rho_{12} = 0$.

Scenario 4: This scenario is equivalent to scenario 3, except that $\rho_{12} \neq 0$.

Each scenario corresponds to L cases of interest. In each case in the list of L cases of interest, a distinct value is assigned to parameter α. The other parameters of the covariance matrix remain constant across all L cases.[2]

Note that the parameter α corresponds to one of the following three quantities: σ_3, ρ_{12}, or ρ_{13}. In scenario 1 we vary σ_3; in scenario 2 we vary ρ_{12}; and in scenarios 3 and 4 we vary ρ_{13} (with $\rho_{13} = \rho_{23}$). Scenario 3 assumes $\rho_{12} = 0$, whereas scenario 4 assumes $\rho_{12} \neq 0$.

In the bottom right corner of Figure 2(b), we indicate the outputs of the simulator for each scenario. Such outputs summarize the simulation of L cases of interest in tables and graphs, including, for example, the following elements:

- $P(error)$ as a function of the number of observations $\mathbf{N}$, with each curve corresponding to a value of d, $d = 2, 3$, and for each value of the parameter α considered;
- Theoretical minimum $P(error)$ as a function of the values of the parameter α considered;
- $P(error)$ as a function of the values of the parameter α considered, with a curve for each value of n considered, $n \in \mathbf{N}$.

The first class of curves above considers both cases with 2 and 3 features, while the latter two classes of curves consider the scenario with 3 features. In these latter two classes, to capture the relevance of the third feature, X_3, we also consider special metrics that indicate the utility of X_3, as detailed in Appendix H.3.

1.6 ILLUSTRATION OF RESULTS AND METHODOLOGY

To illustrate the relationship between sample size, number of features, and the asymptotic error probability of the trained classifier, we considered a simple dataset with either 1 or 2 features. We assumed that the conditional distribution of each sample given its class is a univariate or bivariate Gaussian distribution (as in, for example, (WILLETT; SWASZEK; BLUM, 2000)).

[2] In economics, such an analysis where one parameter is modified while others are kept constant is referred to as an analysis where one parameter is varied while the others are held ceteris paribus.

Table 1 – Error Probabilities for **SVM** (multiplied by 100), $\rho = 0$

		$L_1(n)$,	$L_2(n;\delta)$, 2 features					
n	1 feature	δ=1	δ=2	δ=3	δ=4	δ=5	δ=6	δ=7
2	20.50	14.99	23.29	26.72	28.20	28.93	29.36	29.63
4	18.83	13.91	21.73	24.47	25.62	26.20	26.52	26.73
8	17.63	11.98	18.95	21.02	21.82	22.21	22.42	22.54
16	16.87	10.24	16.28	17.92	18.55	18.86	19.01	19.11
32	16.39	9.18	14.74	16.22	16.78	17.05	17.20	17.29
64	16.15	8.57	13.96	15.39	15.94	16.20	16.34	16.43
129	16.01	8.24	13.57	14.99	15.53	15.79	15.93	16.02
256	15.94	8.06	13.38	14.79	15.33	15.59	15.73	15.82
512	15.90	7.96	13.28	14.69	15.23	15.49	15.63	15.72
1024	15.88	7.91	13.23	14.64	15.18	15.44	15.58	15.67
$n^\star$			12	28	46	70	98	130
Linear Regression for Error Probability: $E(n) = \alpha' + \beta'/n$.								
α'	0.159	0.080	0.132	0.146	0.151	0.154	0.155	0.156
β'	0.142	0.333	0.468	0.522	0.542	0.552	0.556	0.558
$n^\star$			12	30	52	78	106	136

In the bivariate case, we have a correlation coefficient ρ; one of the features has variance 1 and the other has variance δ^2. We observe that for $\rho = 0$, the discriminative power of the second feature decreases as δ increases. The covariance matrix, shared by both classes, is given by

$$\Sigma = \begin{bmatrix} 1 & \rho\delta \\ \rho\delta & \delta^2 \end{bmatrix}.$$

Although this example is intentionally simple, real data can often be approximated by a bivariate Gaussian model. Moreover, the example already serves our purposes: it shows that *(a)* depending on the discriminative power of the second feature, it may be reasonable to ignore it, and *(b)* additional samples may compensate for the absence of a feature (Table 1).

1.7 SIMULATION

1.7.1 Summary of goals

To summarize, we aim to:

1. Compare scenarios where 3 features, X_1, X_2, and X_3, are present against scenarios where only X_1 and X_2 are present.

2. Evaluate the expected classification error as the dataset cardinality n increases.

3. Estimate how the threshold n^* behaves for a given combination of covariance matrix parameters.

1.7.2 Classification Problem

We define a classification problem characterized by:

- A combination of covariance matrix parameters of the Gaussian distribution considered, namely
$$\Sigma = \begin{bmatrix} 1 & \rho\delta \\ \rho\delta & \delta^2 \end{bmatrix}$$
in the bivariate case, and more generally, still in the bivariate case,
$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$
In the three-dimensional case, we have as parameters $\sigma_1, \sigma_2, \sigma_3, \rho_{12}, \rho_{13}, \rho_{23}$, which form the covariance matrix,
$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{bmatrix}.$$
Unless stated otherwise, we focus on the bivariate and three-dimensional cases in this work.

- A set of cardinalities $\mathbf{N} = \{n_1, ..., n_k\}$.

For this classification problem, we generate many synthetic datasets and, for each dataset, compute the separating hyperplane using a linear classifier (**SVM**).

1.7.2.1 What is the Impact of the Number of Observations and Features on Classification Error?

To address the first two goals in Section 1.7.1, we plot two error curves for each case of interest. One curve uses three features (X_1, X_2, X_3), and the other uses only X_1 and X_2. Each curve shows $P(\text{Error})$ as a function of the number of observations n. The threshold $n^\star$ (Table 1) is the n value where the two curves intersect.

1.7.2.2 What is the Impact of Feature Correlation and Variance on Classification Error?

To address the third goal in Section 1.7.1, we proceed as follows. For each of the four scenarios described in Section 1.5.2:

1. Specify a list containing L combinations of parameters characterizing cases to be simulated.

2. Execute the simulation for all L parameter combinations in this list.

3. Interpret the results and connect the interpretations with the geometry of the problem.

1.8 ADDITIONAL DETAILS ABOUT THE CLASSIFICATION PROBLEM CONSIDERED

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 feature vector.

- The response Y can be -1 or $+1$.

- We only consider classifiers from a specific dictionary H (e.g., the set of linear classifiers).

Given a dataset $\mathcal{D}$ with n pairs (x_i, y_i), we choose a classifier $h \in H$. How should we make this choice?

In general, the lower the expected error rate, the better the classifier. Because we work with synthetically generated data, we can evaluate expected performance using the theoretical loss.

Once we select the best classifier for each case of interest, we investigate a potential trade-off between dimensionality and cardinality in the context of a binary classification problem. The idea is to compare two situations, S_1 and S_2, where:

- In situation S_1, we have dimensionality d' and cardinality n'.

- In situation S_2, we have dimensionality d'' and cardinality n''.

- All features present in S_1 are also present in S_2.

If $d' < d''$, can the lower dimensionality of situation S_1 be compensated for by an increase in the dataset cardinality n', i.e., by making $n' > n''$?

Of course, answering this question is not just about comparing dimensionalities and cardinalities. Suppose we consider adding new features to the dataset A to increase its discrimination capacity. Indeed, there are features that, when added, significantly improve the model's discriminative power, while others would be practically useless for this purpose. Thus, it is not just a matter of how many new features are added to the dataset. Depending 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, adding it will be more or less advantageous.

In Table 2, we present a summary of the notation adopted in the remainder of this work.

Table 2 – Notation Table

Variable	Meaning
n	Cardinality being evaluated (each class has $n/2$ observations)
n_k	Maximum cardinality considered in the case of interest
$\mathbf{N}$	Set of cardinalities considered in the case of interest
d	Dimensionality being evaluated
D	Maximum dimensionality considered in the case of interest
(X, Y)	Random vector of features and response
Y	Response, which can be -1 or +1
X	d-dimensional feature vector
$\mathcal{D}$	Dataset (synthetically sampled from Gaussians)
$\boldsymbol{\mu}_+$	Mean vector of the Gaussian distribution for class +1, by default (+1,+1) in the bivariate case and (+1,+1,+1) in the three-dimensional case
$\boldsymbol{\mu}_-$	Mean vector of the Gaussian distribution for class -1, by default (-1,-1) in the bivariate case and (-1,-1,-1) in the three-dimensional case
$\boldsymbol{\mu}$	Mean vector, $\boldsymbol{\mu} = (\boldsymbol{\mu}_+, \boldsymbol{\mu}_-)$
$(\boldsymbol{\sigma}, \boldsymbol{\rho})$	Parameters of the covariance matrix shared between the two classes
Σ	Covariance matrix, shared between the two classes
$(\Sigma, \boldsymbol{\mu})$	Parameters of the Gaussian model
δ^2	Variance of feature X_2, in a special case of the bivariate model
$P(error)$	Error probability of the classifier
$h(x)$	Classification function, or classifier
$L(h(x))$	Loss function (Error) for classifier $h(x)$
$n^\star$	Cardinality threshold beyond which it is advantageous to use feature X_3 given features X_1 and X_2
$\Phi(x)$	CDF of the univariate Gaussian with mean 0 and variance 1, i.e., Cumulative distribution function of the standard normal distribution

1.9 TRAINING EMPIRICAL ERROR AND THEORETICAL ERROR

Below, we introduce the concepts of **Training Empirical Error** and **Theoretical Error**. These concepts will be explored in more detail in Section 2.4 (equations (2.1) and (D.7), respectively).

Training Empirical Error refers to the fraction of misclassified samples, as observed exclusively using the training data. For calculation purposes, we assume the following are known: (a) the training data; (b) the classifier function, empirically obtained from the training data. As seen in equation 2.1

Theoretical Error refers to the fraction of misclassified samples, as predicted by the Gaussian probabilistic model defined by:

- standard deviations $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_D)$
- correlations $\boldsymbol{\rho} = (\rho_{i,j}),\ 1 \leq i < j \leq D$
- means $\mu_+ = [+\mu_1, \ldots, +\mu_D]$ and $\mu_- = [-\mu_1, \ldots, -\mu_D]$

To calculate the **Theoretical Error**, we assume the following are known: (1) the parameters of the Gaussian model ($\boldsymbol{\sigma}$, $\boldsymbol{\rho}$, $\boldsymbol{\mu}$); (2) the equation of the class separation surface, learned empirically from the training data. As seen in equation D.7

1.9.1 Univariate Example

Next, we consider the example illustrated in Figures 3 and 4. In the examples of this section, we consider two univariate normal distributions, both with a standard deviation of $\sigma = 1$ and means $\mu_{red} = -1$ and $\mu_{blue} = +1$. In the terminology of Section 1.8, we refer to the two data classes as corresponding to responses $Y = -1$ and $Y = +1$. In the following example, for visual purposes, red corresponds to -1, and blue corresponds to $+1$.

In Figure 3, we consider a sample containing two observations ($n = 2$), one from each class, where the red and blue observations correspond to the values 0 and $+1$, respectively (marked on the x-axis in Figure 3). In this case, the best classifier learned from the data consists of a classifier that labels the points to the left of $c = 0.5$ as red points, and those to the right as blue. There is nothing more sophisticated that can be learned from such limited data. The **Training Empirical Error** is 0, but the **Theoretical Error** is equal to

$$L_1(0.5) = \frac{1}{2}(1 - \Phi(0.5)) + \frac{1}{2}(1 - \Phi(1.5)) = 0.188$$

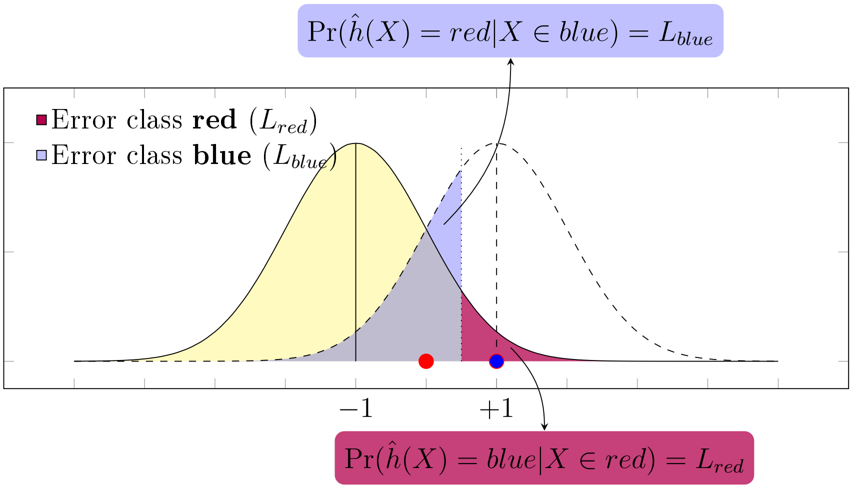


Figure 3 – Binary classification problem, univariate case: the empirically found cut-off point is $c = 0.5$

In Figure 4, we consider a sample containing four observations ($n = 4$), two from each class, with the red and blue observations corresponding to the values $\{-1, +2\}$ and $\{-2, +1\}$, respectively. In this case, the best classifier learned from the data consists of a classifier that labels points to the left of the origin ($c = 0$) as red, and points to the right as blue. The **Training Empirical Error** is 0.5, but the theoretical error, **Theoretical Error**, is equal to

$$L_1(0) = 1 - \Phi(1) = 0.158$$

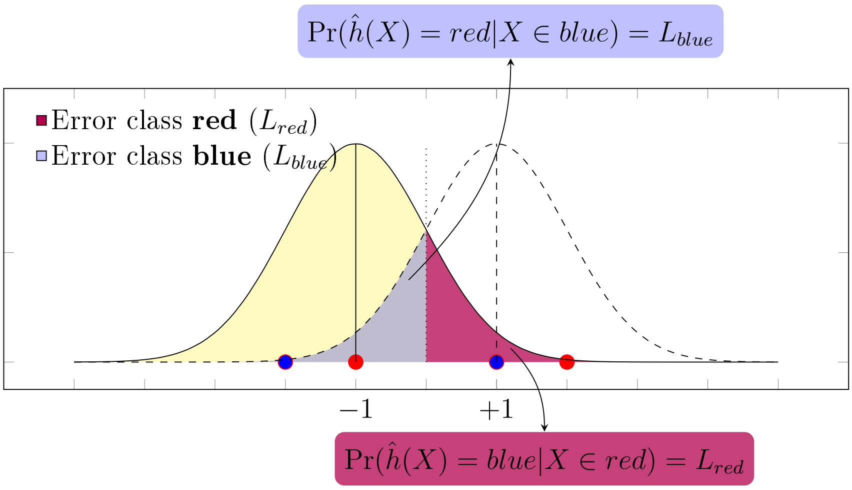
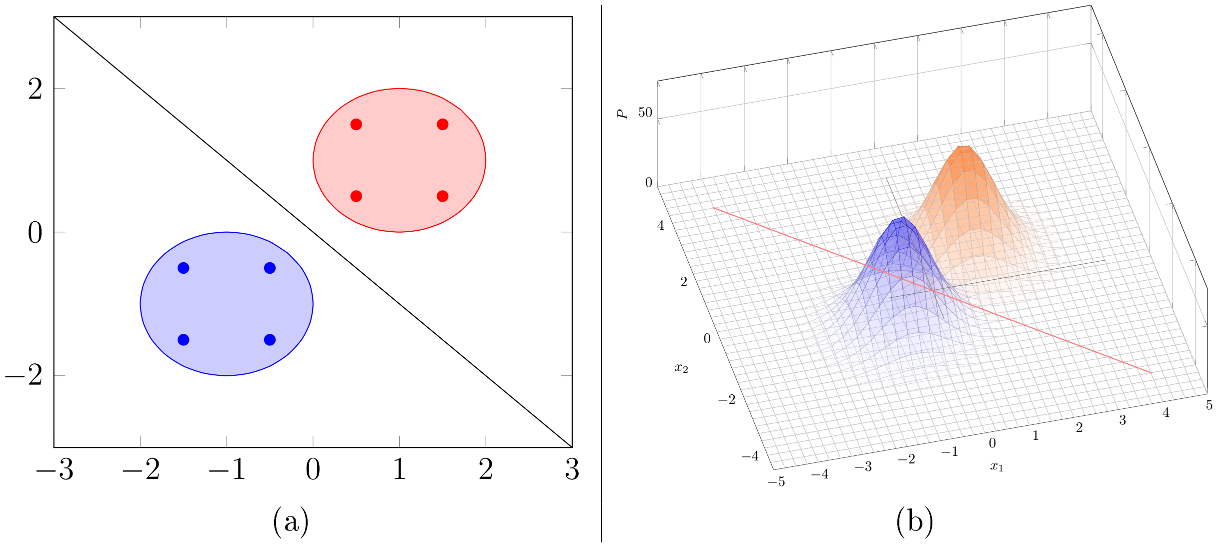


Figure 4 – Binary classification problem, univariate case: the empirically found optimal cut-off point is $c = 0$

Comparing the two figures above, we notice that in the first one, the empirical error is lower, but the theoretical error is higher.

1.9.2 Bivariate Example

As a second example, to illustrate the distinction between empirical error and theoretical error, Figure 5 shows two bivariate Gaussians. Each Gaussian has a covariance matrix equal to the identity, and the means of the Gaussians are (-1,-1) and (+1,+1), for the blue and red Gaussians, respectively. In Figure 5(a), we see 4 observations from each Gaussian, generating a classification empirical error of zero. The best separation line is the line $y = -x$. In Figure 5(b), we observe the underlying Gaussian model, corresponding to a theoretical error greater than zero, calculated in Appendix D.2. One of the purposes of the simulator developed in this work is to evaluate different errors, theoretical and empirical, under different scenarios and identify how the different parameters of the Gaussians affect the errors.



(a) (b)

Figure 5 – Bivariate Gaussian: (a) empirical error equal to zero, considering only generated data; (b) theoretical error greater than zero, considering underlying model.

1.10 CONTRIBUTIONS

Our main contribution is a simulator for evaluating the performance of the **SVM** classification algorithm under a set of Gaussian data. The package is available on PyPI and can be installed via `pip install slacgs`. The source code is available at the following links:

- PyPI package: <https://pypi.org/project/slacgs/>
- Source code: <https://github.com/paulorenatoaz/slacgs>

The development of the simulator is our main contribution, as detailed in Chapter 2. With the simulator, we gathered several interesting observations for bivariate and trivariate Gaussians, and such illustrative results are presented in Chapter 3.

1.11 OUTLINE

The rest of this work is organized as follows. In Chapter 2, we present the implemented simulator. In Chapter 3, we illustrate some results that can be obtained from the simulator. Chapter 4 discusses related work, and Chapter 5 concludes.

We then present a series of additional results in appendices. Appendices A and B provide a contextualization of the problem considered in this work, as well as an introduction to Gaussian distributions, respectively. Then, Appendices C and D discuss types of errors and the calculation of such errors in special cases when the number of samples is large. Still considering a large number of samples, Appendices E and F present additional results for the 2D and 3D cases, respectively. In contrast, Appendix G formally discusses Bayes error for a small sample, with two observations, one in each class. Appendix H presents a heuristic model for understanding the behavior of Bayes error. In particular, in Appendix H.3, we indicate how the proposed simulator generates results to capture the relevance of the third attribute, X_3, considering special metrics that indicate the usefulness of X_3, assuming X_1 and X_2 are given. In other words, we evaluate the gain of considering the trivariate scenario as opposed to the bivariate scenario, using the simulator along with heuristics.

2 SLACGS: A SIMULATOR FOR LOSS ANALYSIS OF CLASSIFIERS USING GAUSSIAN SAMPLES

This chapter presents SLACGS, a simulator built to study classification error when the data come from multivariate Gaussian models. Throughout the chapter, we use SLACGS to answer a practical question introduced earlier: given limited resources, is it better to collect more samples or to add more features?

2.1 OVERVIEW

Figure 6 shows the end-to-end flow.[1]

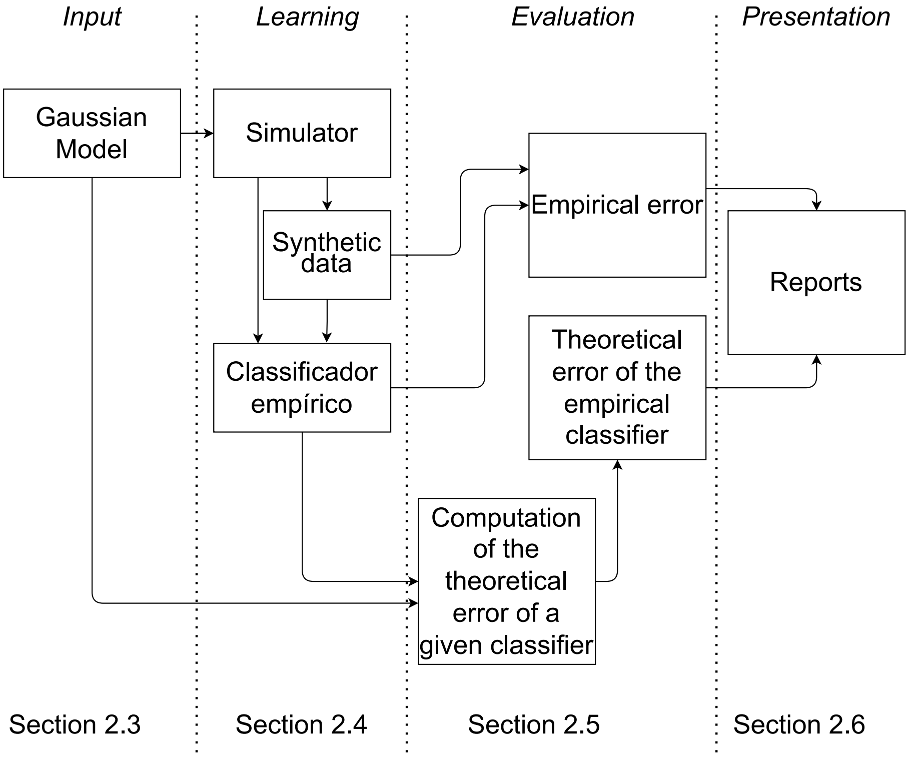


Figure 6 – High-level simulator flow.

Working left to right in Figure 6:

- Input: a Gaussian model that defines two classes via their means and covariance matrices.
- Learning: synthetic datasets are generated incrementally and an **SVM** is trained for each cardinality.

[1] Additional documentation: <https://slacgs.netlify.app/>

- Evaluation: we estimate theoretical and empirical errors for each cardinality and dimensionality.
- Presentation: results are saved as JSON data files and exported as HTML reports, alongside PNG figures and CSV tables, enabling inspection of loss curves across sample sizes and feature dimensions.

2.1.1 Input: Gaussian model

In SLACGS, the two classes are modeled as multivariate Gaussians with symmetric means (by default, $\mu_{(+)} = [+1, \ldots, +1]$ and $\mu_{(-)} = [-1, \ldots, -1]$). The user provides the covariance structure through a compact parameter vector:

- In d dimensions: $[\sigma_1, \ldots, \sigma_d, \rho_{12}, \rho_{13}, \ldots, \rho_{(d-1)d}]$, where the ρ values are listed in upper-triangular order.
- Example (2D): $[\sigma_1, \sigma_2, \rho_{12}]$
- Example (3D): $[\sigma_1, \sigma_2, \sigma_3, \rho_{12}, \rho_{13}, \rho_{23}]$

From this vector, SLACGS constructs the covariance matrix Σ, validates basic consistency conditions (e.g., symmetry and positive definiteness), and defines the cardinalities to simulate (a power-of-two sequence such as $\mathbf{N} = [2, 4, 8, \ldots, 1024]$). This model is then used to generate synthetic training and test datasets across multiple repetitions.

2.1.2 CLI quick-start

SLACGS is driven by a command-line interface (CLI). The examples below mirror the standard demo workflow.

1. Download and install

```
pip install slacgs
slacgs --help
```

2. Predefined experiment scenarios

```
# Run all predefined scenarios
slacgs run-experiment

# Run scenarios 1, 2, 3
slacgs run-experiment --scenarios 1,2,3

# Fast exploratory run
slacgs run-experiment --scenarios 1 --test-mode
```

3 Run a single simulation

```
# 2D
slacgs run-simulation --params "[1,4,0.6]" --test-mode

# 4D (sigmas + correlations)
slacgs run-simulation --params "[1,1,1,2,0,0,0,0,0,0]" --test-mode
```

4 Run custom experiments

```
# Plot 3D ellipsoids for both classes
for samples, color in zip([samples_blue, samples_orange], [blue_color,
↪   orange_color]):
    mean_vector = np.mean(samples, axis=0)[list(combo)]
    sub_cov_matrix = cov_matrix[np.ix_(combo, combo)]

    eigvals, eigvecs = np.linalg.eigh(sub_cov_matrix)
    sorted_indices = np.argsort(eigvals)[::-1]
    eigvals = eigvals[sorted_indices]
    eigvecs = eigvecs[:, sorted_indices]

    for scaling_factor, alpha in zip([1, 4], [1, 0.3]):
        radii = scaling_factor * np.sqrt(eigvals)
        u = np.linspace(0, 2 * np.pi, 100)
        v = np.linspace(0, np.pi, 100)
        x = radii[0] * np.outer(np.cos(u), np.sin(v))
        y = radii[1] * np.outer(np.sin(u), np.sin(v))
        z = radii[2] * np.outer(np.ones_like(u), np.cos(v))

        ellipsoid = np.array([x.flatten(), y.flatten(), z.flatten()]).T
        transformed_ellipsoid = np.dot(eigvecs, ellipsoid.T).T
        transformed_ellipsoid += mean_vector
        transformed_ellipsoid = transformed_ellipsoid.reshape((100, 100, 3))

        ax.plot_surface(
            transformed_ellipsoid[:, :, 0],
            transformed_ellipsoid[:, :, 1],
            transformed_ellipsoid[:, :, 2],
            rstride=4,
            cstride=4,
            color=color,
            alpha=ellipsoid_alpha,
            edgecolor='none'
        )
```

By default, outputs are written under `~/slacgs/output/`. For the complete command reference, run `slacgs -help`.

2.2 CORE COMPONENTS

The simulator architecture consists of three main components: (i) a `Model` class that encodes the Gaussian setup, (ii) a `Simulator` class that orchestrates experiments, and (iii) a reporting module that materializes results in JSON and HTML formats. The CLI provides the primary user interface to these components.

2.2.1 Class Model

The `Model` class describes a two-class multivariate Gaussian setup. Users provide the vector of standard deviations and correlations (for any dimensionality $d \geq 2$). The class validates that the resulting covariance matrix is symmetric and positive definite. It also stores the list of cardinalities $\mathbf{N}$ that the simulator will iterate over, honoring the configuration of the output directory.

2.3 CLASS SIMULATOR

The `Simulator` orchestrates the experiment: generating datasets, training classifiers, and estimating errors.

2.3.1 Elements

A simulator instance holds:

- m: a `Model` object (Section 2.2.1).
- $\mathbf{d}$: dimensionalities to simulate, with $d_i \leq D$, where D is the maximum dimensionality of m.
- $\mathbf{L}$: the loss functions evaluated for each covariance Σ, cardinality $n \in \mathbf{N}$, and dimensionality $d \in \mathbf{d}$, where

  $$\mathbf{L} = [L_{\text{THEORETICAL}}, L_{\text{EMPIRICAL\_TRAIN}}, L_{\text{EMPIRICAL\_TEST}}].$$

  All losses depend on Σ, on $d < D$, and on $n \in \mathbf{N}$. A generic loss can be written as $L_i = L_i(\Sigma, n, d)$ (see Appendix C).

2.3.2 Method run()

Algorithm 1 shows the main loop. For each cardinality n_j, we repeat: (1) generate a dataset, (2) train an SVM for each dimensionality of interest, and (3) update the loss estimates. Lines 5–10 iterate over dimensionalities.

"Algorithm" 1 Simplified `run` method of the `Simulator` class

```
for each cardinality $n_j \in \mathbf{N}$ do
    Let $R(n_j)$ be the number of repetitions to be performed by the simulator
      for cardinality $n_j$
    for  each repetition $r$ from 1 to $R(n_j)$ do
        Generate dataset $\mathcal{D}(\Sigma, r) = \mathcal{D}_{train} \cup \mathcal{D}_{test}$
        for each dimensionality $d_i$ of interest do
            Train an SVM classifier with the training set $\mathcal{D}_{train}$
              in a $d_i$-dimensional space
            for each error type $L_k$ do
                Compute error $L_k(\Sigma, n_j, d_i; r, \mathcal{D})$ and update $\mathbb{E}[L_k(\Sigma, n_j, d_i)]$
            end for
        end for
    end for
    Report $\mathbb{E}[L_k(\Sigma, n_j, d_i)]$, $\forall i, \forall k$
end for
```

The simplified version omits stopping conditions: the simulator can halt early once estimates stabilize. The next subsection includes those checks.

2.3.2.1 Detailed algorithm

Algorithm 2 adds the stopping logic. Line 11 stops iterations for a given cardinality when all loss estimates change less than ϵ between steps. Lines 15–18 extend the cardinality list if needed to find the intersection $n^{\star}$ between error curves for dimensions D and $D-1$.

2.3.2.2 Stopping criteria

The simulation ends only when both conditions hold:

- Intersection found: stop only if, for some $d_{i-1} \leq d_i$, $L_k(\Sigma, n, d_i) < L_k(\Sigma, n, d_{i-1})$ for all k, meaning $n^{\star}$ has been reached.
- Curve convergence: stop only if $L(\Sigma, n_i, d) - L(\Sigma, n_{i-1}, d) < \sqrt{\epsilon}$. Otherwise keep adding repetitions.

2.4 FUNCTIONS IN SIMULATOR.PY: CALCULATING THEORETICAL AND EMPIRICAL ERRORS

2.4.1 Empirical error

The `loss_empirical` function computes misclassification rate for a classifier $\hat{h}^{(\mathcal{D})}$ trained on $\mathcal{D}_{train}$ and evaluated on $\mathcal{D}_{test}$.

"Algorithm" 2 Extended `run` method of the `Simulator` class

```
for each untreated cardinality $n_j \in \mathbf{N}$ do
    Let $R(n_j)$ be the number of repetitions to be performed by the simulator
      for cardinality $n_j$
    for  each repetition $r$ from 1 to $R(n_j)$ do
        Generate dataset $\mathcal{D}(n_j, \Sigma, r) = \mathcal{D}_{train}(n_j, \Sigma, r) \cup \mathcal{D}_{test}(n_j, \Sigma, r)$
          adding elements to $\mathcal{D}(n_{j-1}, \Sigma, r)$
        for each dimensionality $d_i$ of interest do
            Train an SVM classifier with the training set $\mathcal{D}_{train}$
              in a $d_i$-dimensional space
            for each error type $L_k$ do
                Compute error $L_k(\Sigma, n_j, d_i; r, \mathcal{D})$ and update $\mathbb{E}[L_k(\Sigma, n_j, d_i)]$
            end for
        end for
        if all last updates are insignificant then
            STOP analysis of the current cardinality $n_j$ and move to $n_{j+1}$
        end if
    end for
    if there are untreated cardinalities in $\mathbf{N}$ then
        CONTINUE
    else
        if  no crossing between error curves found  then
            $n_{j+1} \leftarrow 2\max(\mathbf{N})$
            $\mathbf{N} \leftarrow \mathbf{N} \cup n_{j+1}$
            CONTINUE the simulation for new cardinality $n_{j+1}$
        else
            Report $\mathbb{E}[L_k(\Sigma, n_j, d_i)]$, $\forall i, \forall j, \forall k$
        end if
    end if
end for
```

- Training empirical error (test set equals train set):

$$\hat{L}(\hat{h}^{(\mathcal{D})}) = \frac{1}{n}\sum_{i=1}^{n} 1(y_i \neq \hat{h}(x_i)). \tag{2.1}$$

- Testing empirical error (independent test set):

$$\hat{L}(\hat{h}^{(\mathcal{D})}, \mathcal{D}') = \frac{1}{n}\sum_{i=1}^{n} 1(y_i \neq \hat{h}(x_i)). \tag{2.2}$$

Algorithm 3 summarizes the calculation.

"Algorithm" 3 Calculation of empirical error

```
procedure LOSS_EMPIRICAL( $\hat{h}^{(\mathcal{D}_{train})}, \mathcal{D}_{test}$ )
    Calculate $\hat{L}(\hat{h}^{(\mathcal{D}_{train})}, \mathcal{D}_{test})$:
        $$\hat{L} \leftarrow \frac{1}{n} \sum_{\forall(x_i,y_i)} 1(y_i \neq \hat{h}^{(\mathcal{D}_{train})}(x_i)), \quad (x_i, y_i) \in \mathcal{D}_{test}$$
    return $\hat{L}$
end procedure
```

2.4.2 Theoretical error

The `loss_theoretical` function estimates misclassification probability using probability theory. Inputs:

- An empirical classifier $\hat{h}$ trained on $\mathcal{D}_{train}$ (size n, dimension d).
- Class means $\mu_{(+)} = [+1, \ldots, +1]$ and $\mu_{(-)} = [-1, \ldots, -1]$.
- Covariance matrix Σ used to generate both classes.

We view the separating hyperplane as $a_1 x_1 + \ldots + a_d x_d = \kappa$, or

$$x_d = \mathbf{w}' \cdot \mathbf{v}$$

with $\tilde{b} = -\kappa/a_d$, $\mathbf{w} = [-a_1/a_d, \ldots, -a_{d-1}/a_d, -1]^T$, and $\mathbf{v} = [x_1, \ldots, x_{d-1}, \tilde{b}]^T$. Algorithm 4 outlines the computation.

"Algorithm" 4 Calculation of theoretical error

```
procedure LOSS_THEORETICAL($\hat{h}, \Sigma$)
    Define $\mathbf{a} \leftarrow$ coefficients of the normal vector to the separating hyperplane of $\hat{h}$
    Define $\kappa \leftarrow$ intercept term of the separating hyperplane of $\hat{h}$
    Define $\tilde{b} \leftarrow -\kappa/a_d$
    Define $\mathbf{w} \leftarrow \begin{bmatrix} -a_1/a_d \\ \vdots \\ -a_{d-1}/a_d \\ 1 \end{bmatrix}$
    return h_error_rate($\tilde{b}, \mathbf{w}, \Sigma$)
end procedure
```

2.5 CLASS REPORT

SLACGS separates data capture from presentation. A reporting module consumes a `Simulator` instance, materializes intermediate data structures (e.g., aggregated loss ta-

"Algorithm" 5 Calculating the error probability for a linear classifier

```
procedure H_ERROR_RATE($\tilde{b}, \mathbf{w}, \Sigma$)
    Define $\boldsymbol{\lambda} \leftarrow$ lower triangular factor from the Cholesky decomposition of matrix $\Sigma$
    Define $\Delta$:
        $\tilde{\boldsymbol{\delta}} \leftarrow \boldsymbol{\lambda}^T \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_{d-1} \\ -1 \end{bmatrix} = \begin{bmatrix} \tilde{\delta}_1 \\ \vdots \\ \tilde{\delta}_d \end{bmatrix}$    (2.3)
        $\Delta \leftarrow \sum_{i=1}^{d} \tilde{\delta}_i^2$    (2.4)
    Define Distance(+) and Distance(-):
        $\text{Distance}(+) \leftarrow \frac{|w_1 + \ldots + w_{d-1} - 1 - \tilde{b}|}{\sqrt{\Delta}}$
        $\text{Distance}(-) \leftarrow \frac{|w_1 + \ldots + w_{d-1} - 1 + \tilde{b}|}{\sqrt{\Delta}}$
    Define $P(\text{Error}(+))$ and $P(\text{Error}(-))$ using the standard normal CDF:
        $P(\text{Error}(+)) \leftarrow 1 - \Phi(\text{Distance}(+))$
        $P(\text{Error}(-)) \leftarrow 1 - \Phi(\text{Distance}(-))$
    Calculate the final error probability:
        $P(\text{Error}) \leftarrow \frac{P(\text{Error}(+)) + P(\text{Error}(-))}{2}$
    return $P(\text{Error})$
end procedure
```

bles), and produces both machine-readable outputs (JSON) and human-readable outputs (HTML). Key stored elements include:

- Iterations and maximum iterations per dimensionality.
- Loss values per dimensionality and error type, plus Bayes loss.
- Distances and geometry diagnostics derived from the covariance structure.
- Runtime summaries and metadata tags for reproducibility.
- Rendered figures of the error curves and data plots.

Results are presented in the next chapter.

2.6 CHALLENGES

Key implementation challenges and how we addressed them:

- **Smooth error curves**: cumulative data generation across cardinalities keeps curves stable and reproducible with fixed seeds.
- **General theoretical error**: a dimension-agnostic formula (Algorithm 5) replaces earlier dimensional limits; derivations appear in Appendix D.4.
- **Stopping rules**: early stop per cardinality when loss updates fall below ϵ; extend **N** only when intersections are unresolved.
- **Bayes error without a closed form**: for $d > 3$, approximate Bayes error empirically at the largest cardinality and comparing both empirical error estimations (Section 2.4.1).
- **Reporting pipeline**: distinct data and presentation stages produce JSON plus HTML artifacts that adapt to dimensionality and cardinality grids.

2.7 3D DATA VISUALIZATION

We plot both the separating hyperplane and class ellipsoids using eigenstructure of Σ.

- **Hyperplane**: axes for the two largest standard deviations define a mesh; the third axis is computed from the SVM coefficients.
- **Ellipsoids**: eigenvalues/eigenvectors of the covariance submatrix set radii and orientation. Sorting eigenvalues highlights the principal axes.

The code snippets below illustrate the approach used in `Model.plot_data_3d_3by3`.

```
# get sorted indices of standard deviations in descending order
sorted_indices = np.argsort(sigmas)[::-1]

hyperplane_bound = max(sigmas) * 5
axes = [None, None, None]

axes[sorted_indices[0]], axes[sorted_indices[1]] = np.meshgrid(
    np.linspace(-hyperplane_bound, hyperplane_bound, 50),
    np.linspace(-hyperplane_bound, hyperplane_bound, 50))

axes[sorted_indices[2]] = (
    -clf.intercept_[0]
    - clf.coef_[0][sorted_indices[0]] * axes[sorted_indices[0]]
    - clf.coef_[0][sorted_indices[1]] * axes[sorted_indices[1]]
) / clf.coef_[0][sorted_indices[2]]

ax.plot_surface(axes[0], axes[1], axes[2], color='gray', alpha=0.15, zorder=0)
```

```
# Plot 3D ellipsoids for both classes
for samples, color in zip([samples_blue, samples_orange], [blue_color,
↪   orange_color]):
    mean_vector = np.mean(samples, axis=0)[list(combo)]
    sub_cov_matrix = cov_matrix[np.ix_(combo, combo)]

    eigvals, eigvecs = np.linalg.eigh(sub_cov_matrix)
    sorted_indices = np.argsort(eigvals)[::-1]
    eigvals = eigvals[sorted_indices]
    eigvecs = eigvecs[:, sorted_indices]

    for scaling_factor, alpha in zip([1, 4], [1, 0.3]):
        radii = scaling_factor * np.sqrt(eigvals)
        u = np.linspace(0, 2 * np.pi, 100)
        v = np.linspace(0, np.pi, 100)
        x = radii[0] * np.outer(np.cos(u), np.sin(v))
        y = radii[1] * np.outer(np.sin(u), np.sin(v))
        z = radii[2] * np.outer(np.ones_like(u), np.cos(v))

        ellipsoid = np.array([x.flatten(), y.flatten(), z.flatten()]).T
        transformed_ellipsoid = np.dot(eigvecs, ellipsoid.T).T
        transformed_ellipsoid += mean_vector
        transformed_ellipsoid = transformed_ellipsoid.reshape((100, 100, 3))

        ax.plot_surface(
            transformed_ellipsoid[:, :, 0],
            transformed_ellipsoid[:, :, 1],
            transformed_ellipsoid[:, :, 2],
            rstride=4,
            cstride=4,
            color=color,
            alpha=ellipsoid_alpha,
            edgecolor='none'
        )
```

3 RESULTS

In this section, we present numerical results obtained with the constructed simulator.

3.1 WHEN DOES X_3 CONTRIBUTE TO INCREASING THE DISCRIMINATIVE POWER OF (X_1, X_2)? VISUALIZING CLASSIFICATION INSTANCES

Given a classification problem with the characteristics we are analyzing, i.e., two tri-normal distributions centered respectively at $\mu_{(+)} = (+1, +1, +1)$ and $\mu_{(-)} = (-1, -1, -1)$, both with covariance matrices equal to:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{bmatrix} \tag{3.1}$$

We know that, depending on the values of the six parameters (variances and correlations) that define this matrix Σ, there is a threshold n^* for the number of observations, above which the presence of attribute X_3 becomes advantageous, given that the other two attributes X_1 and X_2 are already present. For a small number of observations, using all three attributes may lead to *overfitting*, and it is advantageous to use only two attributes. As the number of observations increases, it becomes worthwhile to use the third attribute, as the risk of *overfitting* ceases to exist.

Question: Given a matrix Σ, characterized by a particular combination of the six parameters σ_1, σ_2, σ_3, ρ_{12}, ρ_{13}, ρ_{23} that define it, how can we evaluate the potential additional contribution of X_3 to the discrimination between the two groups, once X_1 and X_2 are already present?

To analyze this question, four scenarios were created, in each of which only three of the six parameters can be chosen freely. In each of these four scenarios, given the matrix Σ, the quotient Q can be calculated,

$$Q = \frac{R_2}{R_3}, \tag{3.2}$$

where R_3 and R_2 are the Bayes risks (Bayes error) related, respectively, to the situation in which all three attributes are present and the situation in which only X_1 and X_2 are present. Supposedly, the larger the quotient R_2/R_3, the smaller n^* should be.

3.1.1 Visualizing the Ellipsoids

Visualizing the ellipsoids corresponding to the two tri-normal distributions we want to separate is very useful for gaining intuition about the problem. Our simulator allows us to visualize such ellipsoids.

Figures 7 and 8 illustrate the ellipsoids in two different situations:

- In Figure 7, we have $\rho_{12} < 0$. As can be seen in the figure, this facilitates the classification task. After all, since the centroids are $(+1, +1, +1)$ and $(-1, -1, -1)$, the main axis of the ellipsoids aligns closely with the direction of the line passing through the two centroids;
- In Figure 8, we have $\rho_{12} > 0$. As can be seen in the figure, this makes the classification task more difficult. In the extreme case where $\rho_{12} = 1$, the main axis of the ellipsoids becomes perpendicular to the direction of the line passing through the two centroids.

With the visualization in Figures 7 and 8, it becomes easy to see the impact of ρ_{12} on the classification difficulty, illustrating the usefulness of the simulator (see also Appendix H.1).

3.1.2 Summary

In summary, we know that the level surfaces corresponding to a given tri-normal distribution are ellipsoids centered at their respective centroid (mean vector). Therefore, if the two centroids are $(+1, +1, +1)$ and $(-1, -1, -1)$, the closer the direction of the main axis of these ellipsoids is to the direction of the line passing through these two centroids, the greater the intersection between the point clouds relative to the two groups, i.e., the greater the expected probability of classification error if all three attributes are present.

The proposed simulator helps to visualize these observations in concrete scenarios. Note that to draw the hyperplanes and ellipsoids in Figures 7 and 8, we used the ideas presented in Section 2.7. In particular, to draw the hyperplanes, we first constructed a mesh with the axes of the largest and second-largest standard deviations. This allows a clear visualization of the hyperplane separating the orange and blue ellipsoids, both in Figure 7 and Figure 8. Projections in the planes $X_1 \times X_2$, $X_1 \times X_3$, and $X_2 \times X_3$ are also presented in 2D in the figures.

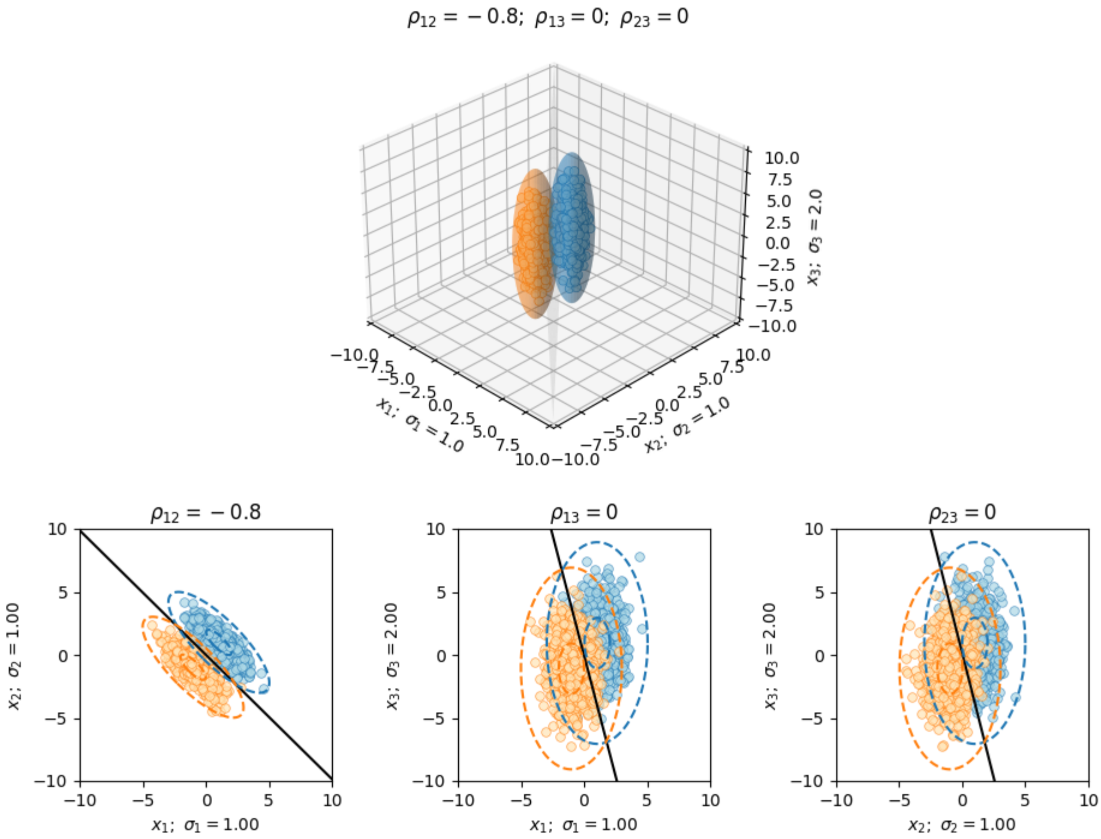


Figure 7 – Visualizing Classification Instance: $\boldsymbol{\sigma} = [1, 1, 2], \boldsymbol{\rho} = [-0.8, 0, 0]$; $n = 1024$

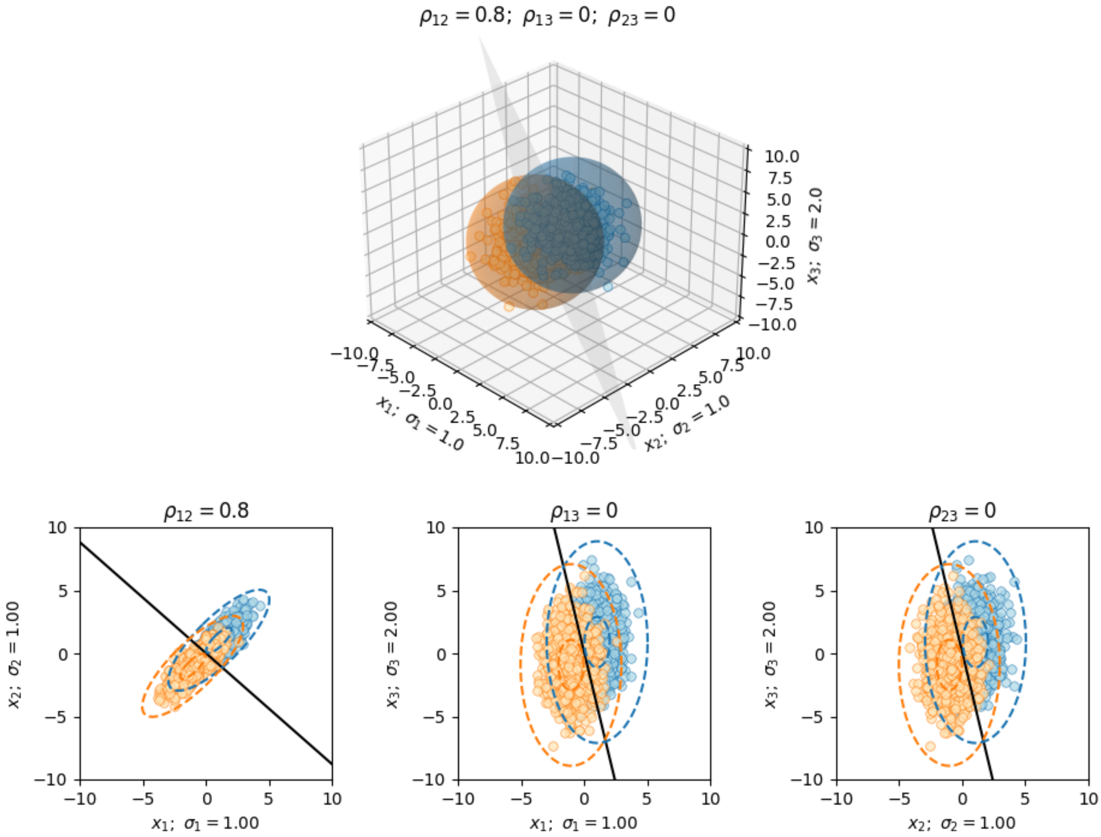


Figure 8 – Visualizing Classification Instance: $\boldsymbol{\sigma} = [1, 1, 2]$, $\boldsymbol{\rho} = [0.8, 0, 0]$; $n = 1024$

3.2 WHAT IS THE IMPACT OF THE NUMBER OF OBSERVATIONS? REPORT PRODUCED BY A SIMULATION CASE

Below, we present examples of results for a simulation case where:

- $\mu_{(+)} = (+1; +1; +1)$ and $\mu_{(-)} = (-1; -1; -1)$
- $\boldsymbol{\sigma} = (1, 1, 2)$
- $\boldsymbol{\rho} = (-0.4; 0; 0)$
- $\mathbf{N} = (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024)$

To recap, the first item refers to the centroids of the classes, the second and third items refer to the parameters of the shared covariance matrix between the classes, and the last item refers to the number of observations in each classification instance. Also recall that a simulation case contains several classification instances, and the goal of a simulation case is to evaluate the impact of the number of observations on the error.

3.2.1 Graphs

We can visualize the error curves grouped by type of error (Figures 9, 10, and 11) or by the number of features (Figures 12, 13, and 14).

In the graphs where the curves are grouped by type of error, we can observe that:

1. In Figure 9, the theoretical error curves for 2 and 3 features intersect at a single point. This point corresponds to the value of n^* discussed in the previous section. When $n \geq n^*$, it becomes advantageous to use the third feature.
2. In Figure 10, the same observations made above continue to hold when analyzing the empirical test error.
3. In Figure 11, we see that the empirical training error behaves differently from those discussed above. In particular, the empirical training error starts at 0 when we have only one observation on each side of the separating hyperplane. That is, for a dataset with cardinality $n = 2$ (one observation on each side of the separating hyperplane), the empirical training error rate is zero. The error grows as the cardinality n increases, and asymptotically tends to the Bayes error. Note that there is no intersection between the empirical training error curves presented in Figure 11.

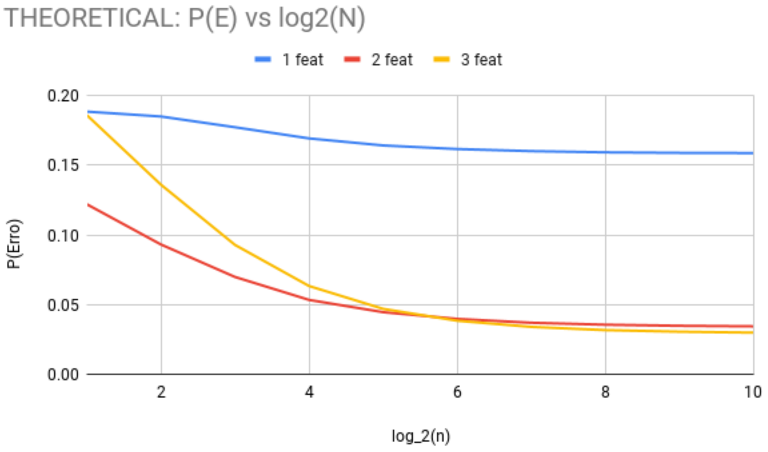


Figure 9 – Theoretical error as a function of the number of observations

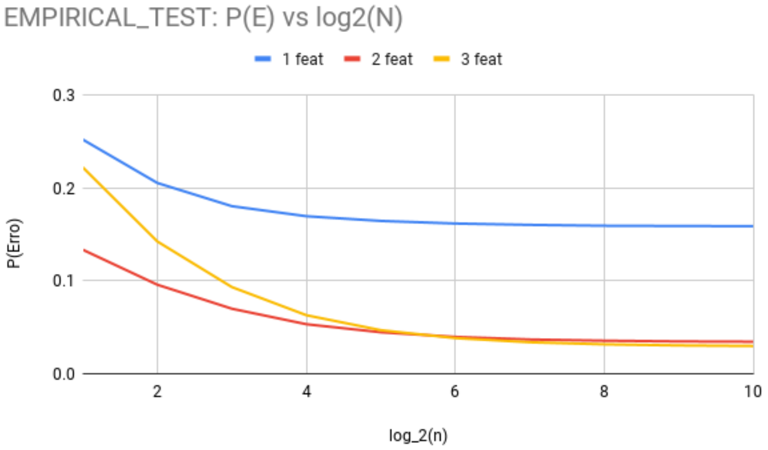


Figure 10 – Empirical test error as a function of the number of observations

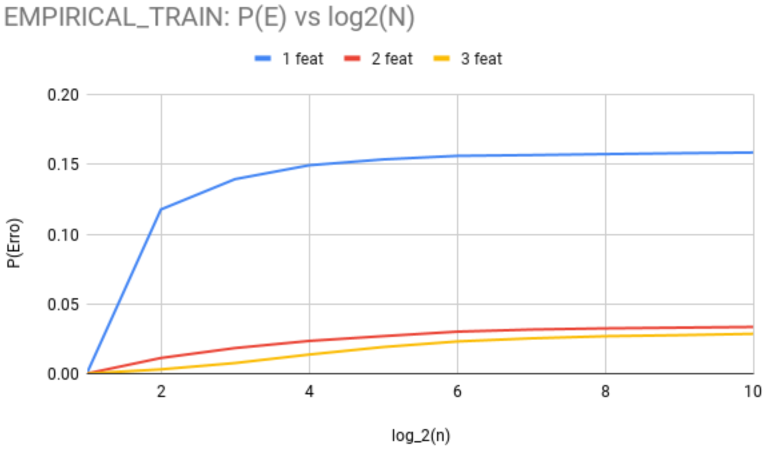


Figure 11 – Empirical training error as a function of the number of observations

Next, we discuss Figures 12, 13, and 14. In these graphs, the curves are grouped by the number of features. We can observe that in all cases, the errors converge to the theoretical Bayes error. Additionally, we note that the empirical test error is always greater than the theoretical error, which in turn is greater than the empirical training error.

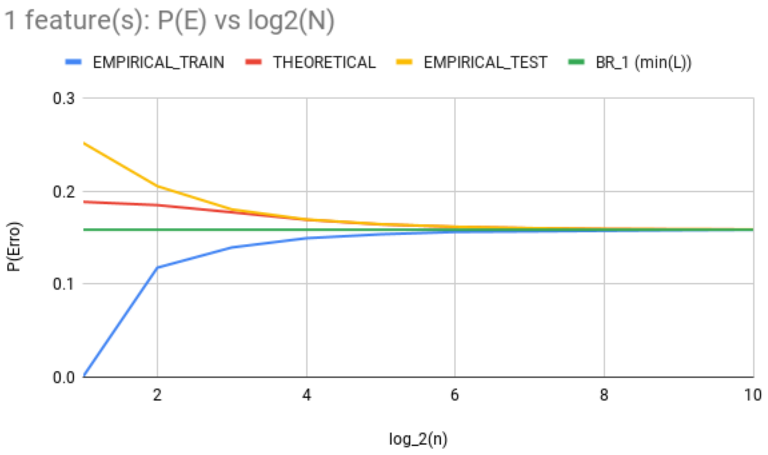


Figure 12 – Error as a function of the number of observations, with 1 feature present

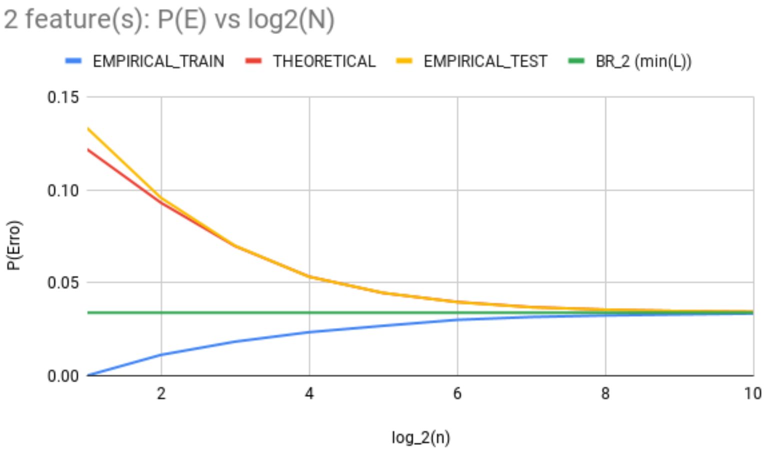


Figure 13 – Error as a function of the number of observations, with 2 features present

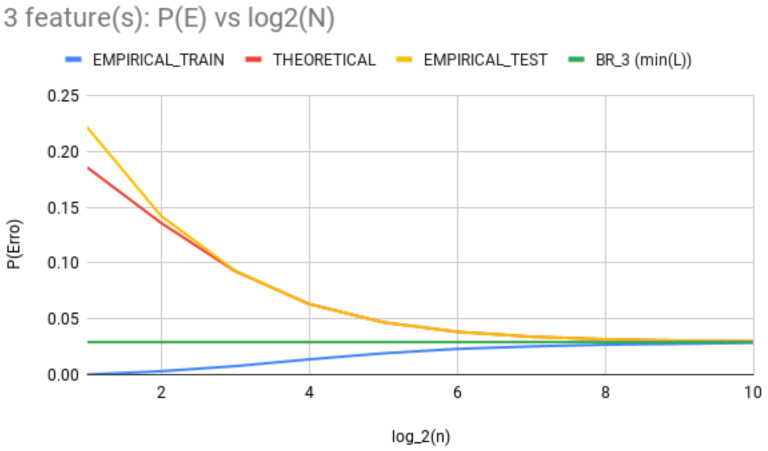


Figure 14 – Error as a function of the number of observations, with 3 features present

3.2.2 Visualizations

At the end of a simulation case, we obtain images (*.png*) containing visualizations of the samples considered. In particular, we generate images for all the dimensions and cardinalities simulated, together with graphs containing the error rate curves. Additionally, we also have an animation (*.gif*) containing all the produced images. Below, we see examples of visualizations for the cardinalities $n = 2$, $n = 4$, $n = 64$, and $n = 1024$, in Figures 15, 16, 17, and 18, respectively.

Some observations about the obtained visualizations:

1. At the top of the visualizations, we have the ellipsoids and separating hyperplane, produced as described in Section 2.7.

2. Next, we see the 2D projections, illustrating the separation between classes through separating lines.

3. In the row with the graphs titled "Feature 1," "Feature 2," and "Feature 3," we see the PDFs of the three features considered. The separating line is marked as a black line. As the cardinality n increases, the independent term in the equation of the separating hyperplane (also referred to as bias b) approaches zero, and the separating plane approaches the origin, as proposed by the analytical model described in Appendix D.

4. Finally, in the last row of the figures, we have the error as a function of the number of observations. For $n = 2$, the graphs are empty. For $n = 4$, the graphs include $n = 2$ and $n = 4$. For $n = 64$, the graphs include $n = 2$, $n = 4$, and $n = 64$, and mark the intersection points between the curves with 2 and 3 features. Finally, for $n = 1024$, we see the complete error curves. This evolution can also be visualized through an animation (animated gif), where the curve is gradually filled in as we consider larger values for n.

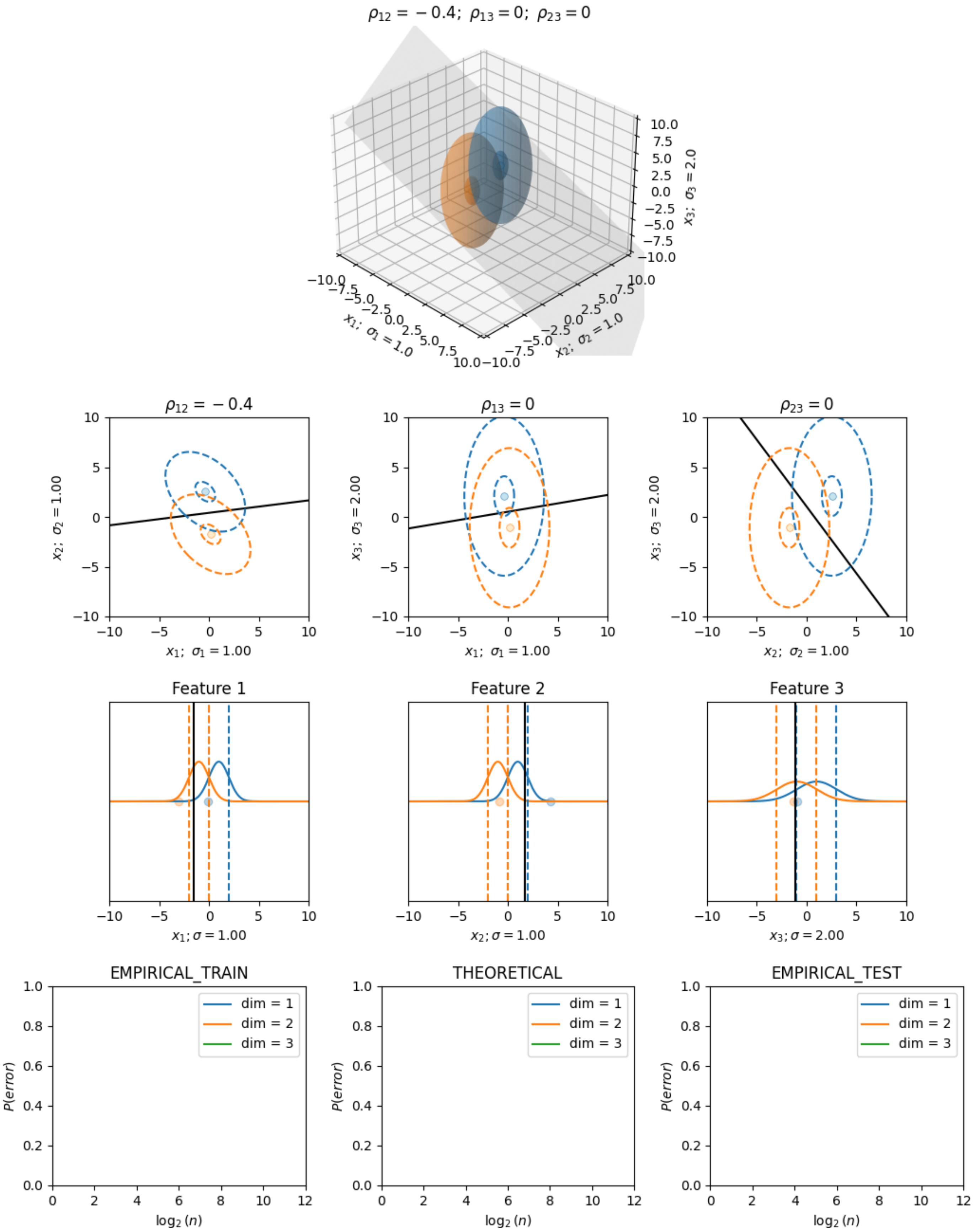


Figure 15 – Visualization of the ellipsoids and separating hyperplane, for $n = 2$

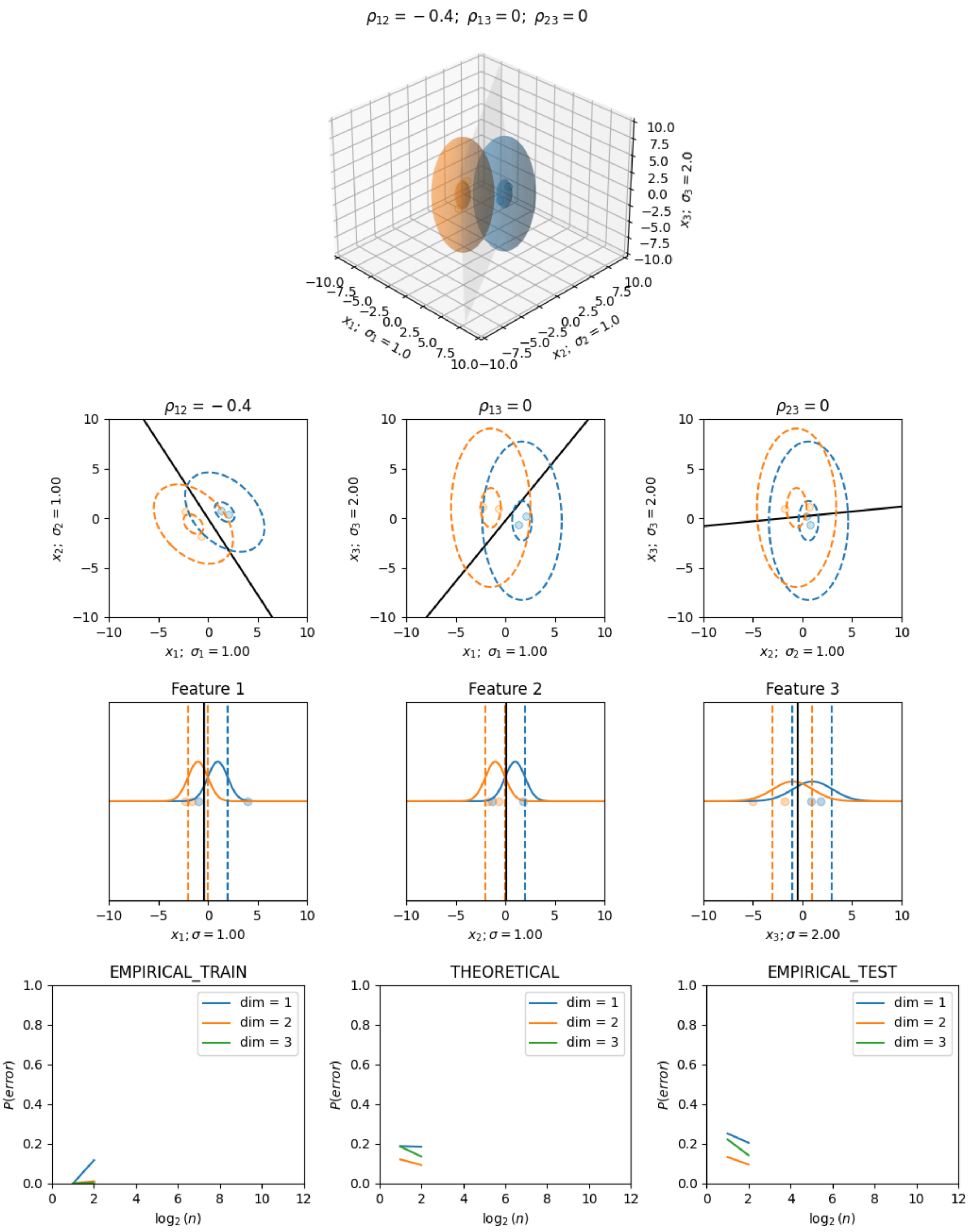


Figure 16 – Visualization of the ellipsoids and separating hyperplane, for $n = 4$

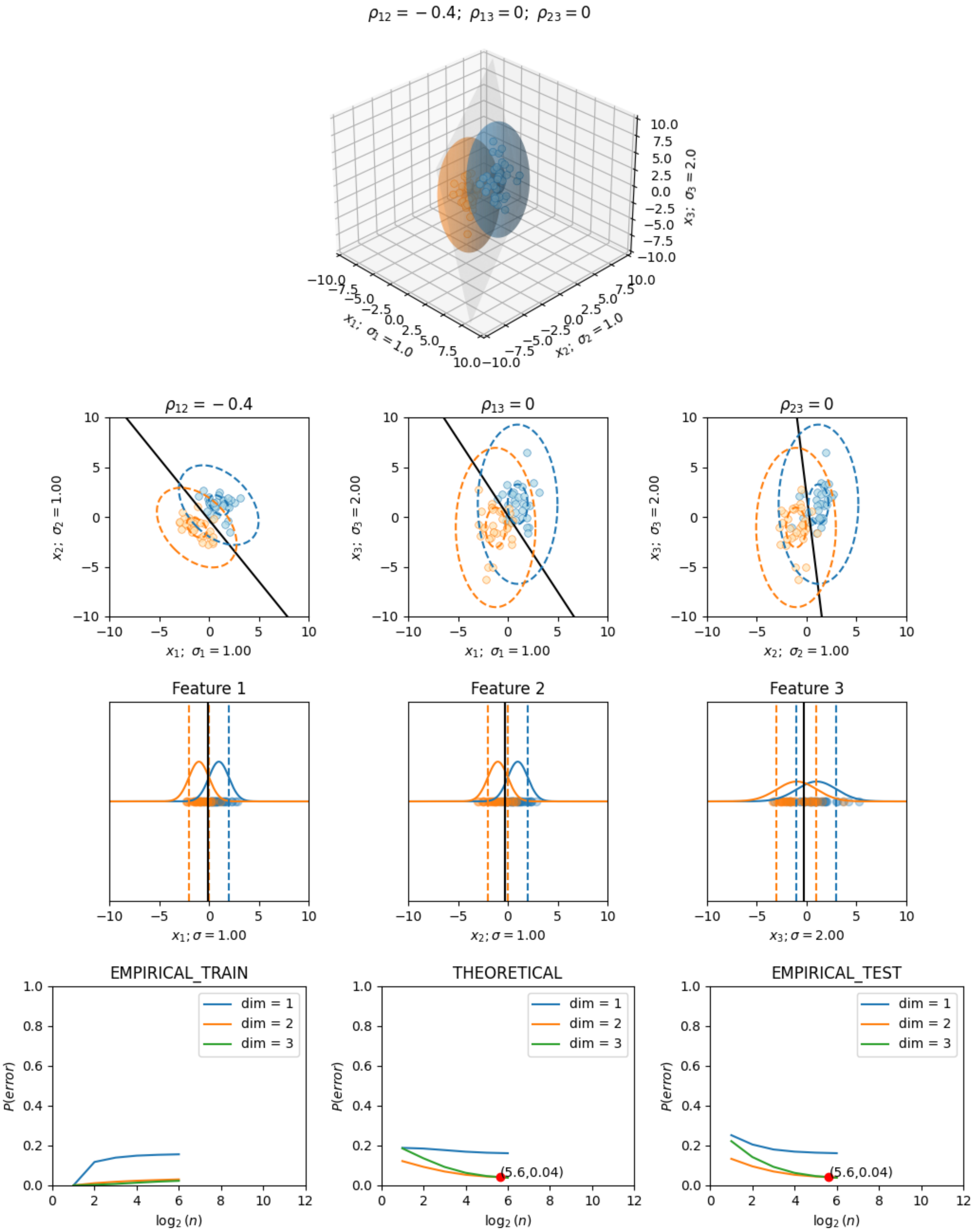


Figure 17 – Visualization of the ellipsoids and separating hyperplane, for $n = 64$

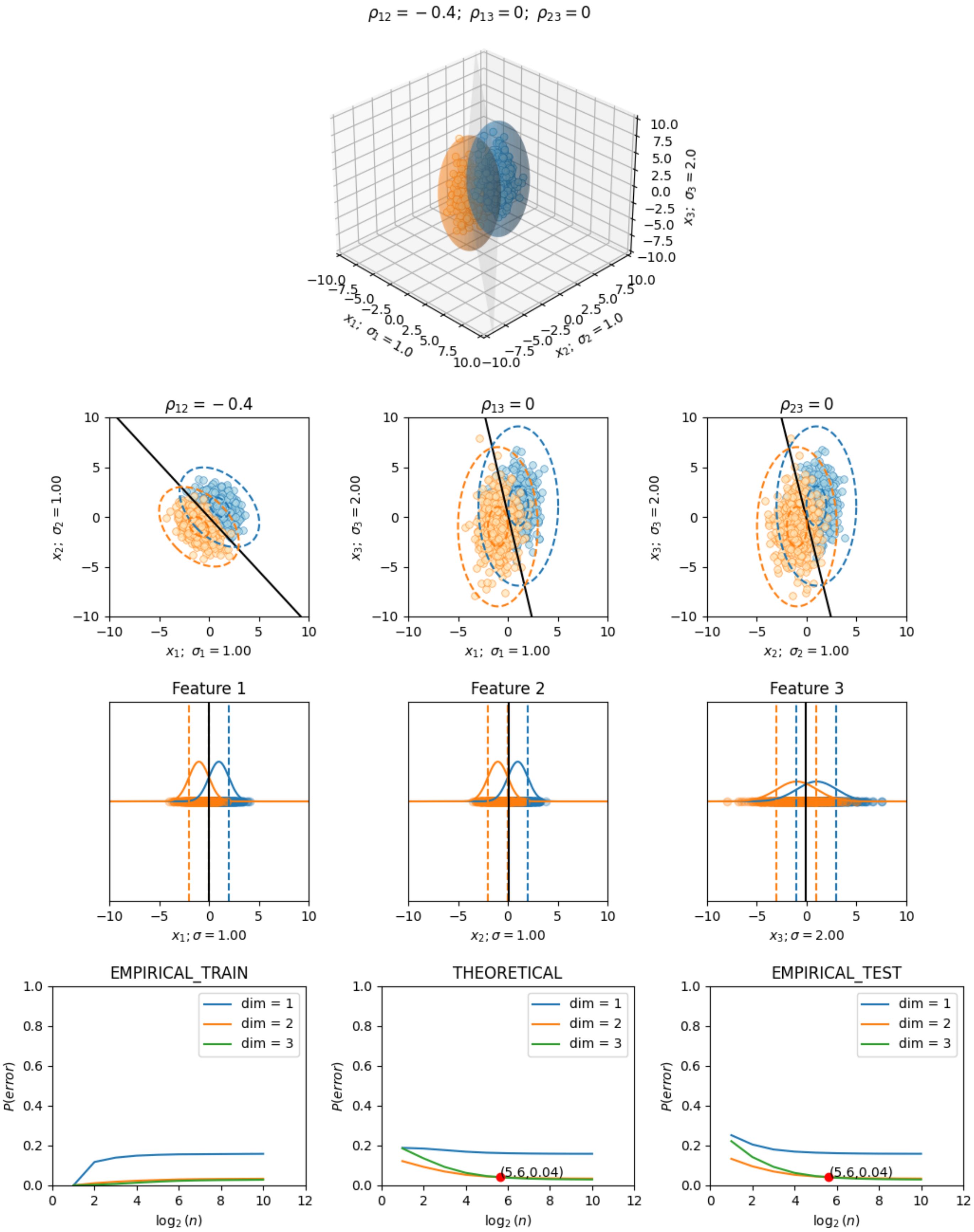


Figure 18 – Visualization of the ellipsoids and separating hyperplane, for $n = 1024$

3.3 WHAT IS THE IMPACT OF THE COVARIANCE MATRIX PARAMETERS? REPORT PRODUCED BY A SCENARIO CONTAINING MULTIPLE SIMULATION CASES

Next, we evaluate the impact of ρ_{12} on the classification problem. This evaluation is within the scope of **Scenario 2**.

In particular, we vary ρ_{12} between -0.8 and 0.2, in steps of 0.1, i.e., $\rho_{12} \in \{-0.8, -0.7, \ldots, 0.2\}$.

Here is the link to view, in the form of an animation, the impact of varying ρ_{12} on the classification model and the separating plane: Scenario 2[1]

3.3.1 What is the Impact of ρ_{12} on the Usefulness of X_3?

We consider the usefulness of feature X_3 for classification. For this, we use the quotient Q introduced in (3.2). Empirically, it is found that the larger Q, the more useful feature X_3 is. However, calculating Q requires simulation rounds. We then ask the following question: is it possible to find an approximation for Q that is easier to obtain and that allows us to determine the usefulness of X_3? In Appendix H, we present a proposal, denoted as d_2/d_3. Appendix H.4 provides additional discussion on the topic.

Figure 19 illustrates the relationship between d_2/d_3 and Q, indicating a linear relationship between them.

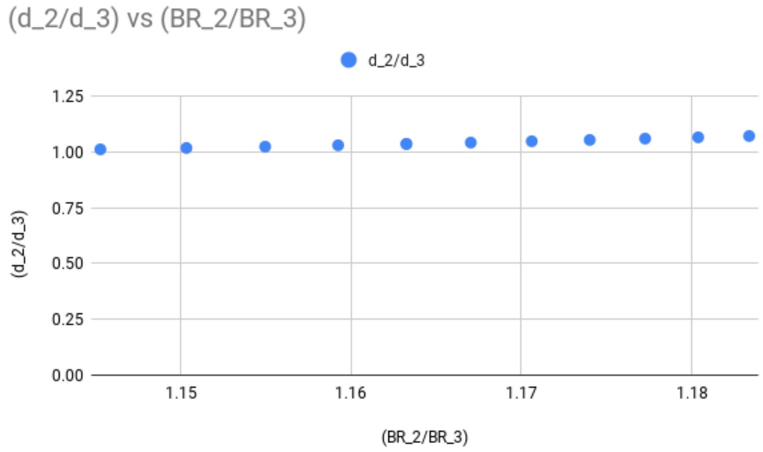


Figure 19 – Relationship between indicators of the usefulness of X_3. The indicators are d_2/d_3 and Q, corresponding to the ratio of Bayes risks with 2 and 3 features. The figure indicates a linear relationship between the two indicators, illustrating another application of the constructed simulator.

[1] <https://drive.google.com/file/d/18xUciYIv-3JguIMDvMbt9XVfJrRpZz0v/view?usp=drive_link>

Figure 20 supports the above argument. It shows the threshold n^* as a function of ρ_{12} in Scenario 2. The higher ρ_{12}, the lower n^*, and the greater the usefulness of X_3.

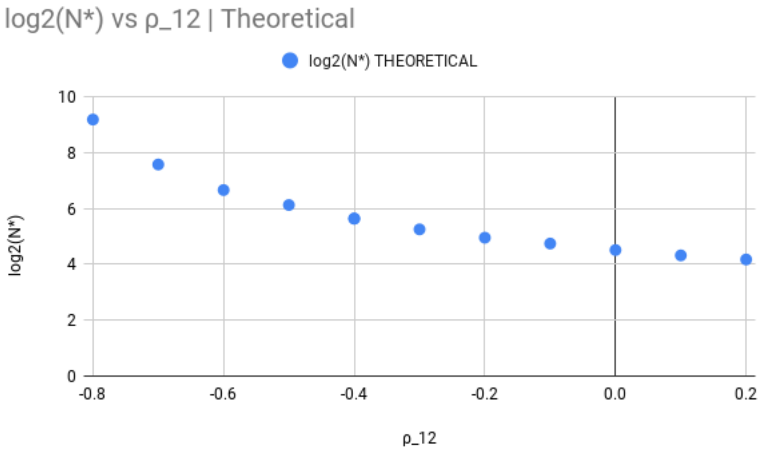


Figure 20 – Threshold n^* as a function of ρ_{12} in Scenario 2: the higher ρ_{12}, the lower n^*, and the greater the usefulness of X_3

3.3.2 Additional Animations

Click the links below to access additional animations, illustrating the other scenarios listed in Appendix F.2:

Scenario 1

Scenario 2

Scenario 3

Scenario 4

4 RELATED WORK

The analytical models presented in this work were based on the doctoral thesis of Prof. João I. Pinheiro (PINHEIRO, 2021). Our main contribution consists of building a simulator to experimentally investigate how different parameters impact linear classifiers on Gaussian samples. While (PINHEIRO, 2021) focused on the 1D and 2D cases, our work allows extrapolating the results to the 3D case.

4.1 LEARNING WITH FEW OBSERVATIONS

Training classifiers with few samples is a topic that has recently gained attention from the remote sensing community (ZHANG et al., 2021; HE et al., 2021), Large Language Models (LLMs) (BROWN et al., 2020), among others (HANCZAR; DOUGHERTY, 2013; MILLER; MATSAKIS; VIOLA, 2000; WANG et al., 2020).

Detecting Change Points. Consider a system that implements an algorithm for detecting change points aimed at tracking abrupt changes in the environment. Once a change point occurs, we may have a few samples from each of the classes representing our target of interest. Training a classifier immediately after such a change point is then restricted to small samples (few observations) and a possibly limited number of features (due to sensor network constraints) (NOH; RAJAGOPAL; KIREMIDJIAN, 2013). Although significant effort has already been made to investigate how the number of samples and the number of features affect classifier accuracy (HANCZAR; DOUGHERTY, 2013; NOH; RAJAGOPAL; KIREMIDJIAN, 2013), the relationship between these two elements still presents open analytical and practical problems.

Bias vs Variance. Another recent line of research refers to the study of the so-called "double descent" effect (DENG; KAMMOUN; THRAMPOULIDIS, 2021; NAKKIRAN et al., 2020; LOOG; VIERING; MEY, 2019; D'ASCOLI; SAGUN; BIROLI, 2020; BELKIN et al., 2019). The classical literature on the relationship between bias and variance suggests that the error probability decreases as the number of parameters increases and then rises again due to "overfitting." The "double descent" literature extends this idea, pointing out that if the number of parameters continues to increase, the error reaches peaks around the point where the model capacity (e.g., measured by the number of features) approximately equals the number of samples (number of observations), and then it falls again. In other words, when systems become complex enough to only fit the training examples, the error increases (due to "overfitting"), but increasing the number of parameters beyond this limit reduces the error again. The isotropic case with $\rho = 0$ and $\delta = 1$ is discussed in (DENG; KAMMOUN; THRAMPOULIDIS, 2021).

In this work, we consider the non-isotropic case and show that reducing the number of features has a regularization effect, naturally leading to better generalization ability, which is critical when we have a small number of observations.

4.2 SIMULATION OF CLASSIFIERS

Similar to what we do here, in (HUA et al., 2005) the authors also deal with binary classification through a simulation-based approach.

Our simulations address specific examples, and we consider what happens to the error probability as the cardinality increases, considering from 1 to 3 features, and Gaussian samples.

On the other hand, in (HUA et al., 2005), the authors consider a much broader combination of probabilistic models and statistical procedures. They show that, under suitable conditions, for each dataset cardinality n, there is an optimal number of features to be used in the classifier to minimize its error probability.

We believe that the simulator proposed here could be used to reproduce some of the results presented in (HUA et al., 2005) and extend them to the specific case where the samples are Gaussian, taking into account how different parameters of the Gaussian model impact the error probability. Such analysis is beyond the scope of (HUA et al., 2005).

4.3 ERROR PROBABILITY AS A FUNCTION OF THE NUMBER OF FEATURES AND OBSERVATIONS

In (HUGHES, 1968), Hughes analyzes the classification of discrete features, presenting some results similar to (HUA et al., 2005). Instead of considering the dimensionality d, referring to the number of features, in (HUGHES, 1968) the concept of "complexity of the measurement pattern" c is introduced, namely the total number of possible values of the feature vector. The author states that for each dataset cardinality n, there is an ideal c for which the classifier's error probability is minimized. Thus, by replacing d with c, the conclusions of (HUGHES, 1968) and (HUA et al., 2005) are analogous to each other.

Both our work and (HUA et al., 2005) and (HUGHES, 1968) point out that, due to the occurrence of *overfitting*, if the dataset cardinality is relatively small, a high dimensionality can substantially impair the expected classifier performance.

Our work suggests that, in many situations, there may be a trade-off between samples and features regarding minimizing the error probability. In other words, an eventual scarcity of features could be compensated by increasing the sample size. This generally tends to happen when the new features that could eventually be added do not have as high a discriminative power as those already present in the model. On the other hand, an eventual scarcity of samples can also be compensated by adding new features to the dataset. This requires the availability of new features whose inclusion significantly improves the model's overall discriminative power.

5 CONCLUSION AND FUTURE WORK

In this work, we considered a binary classification problem involving univariate, bivariate, and trivariate Gaussians. In particular, we presented a simulator to evaluate how the classification error varies as a function of different problem parameters.

Among the applications of the proposed simulator, we analyzed how the cardinality n of the dataset, along with the values of the parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\rho}$, simultaneously influence the decision to use an additional feature in defining the classifier, with the goal of minimizing its error probability. Using the simulator, we verified the existence of a threshold for the cardinality n, below which it is preferable to forego an additional feature.

In this work, we explored 1, 2, and 3-dimensional spaces, but with the developed simulator, we can consider higher dimensions. To illustrate such an extension, we produced Figures 21 and 22. In future work, we intend to assess whether our conclusions about the threshold for n can be extended to higher-dimensional spaces. We also intend to experiment with the trade-off between features and samples in a real environment, for example, involving a wireless sensor network in a test setting. In this case, we aim to verify under what conditions it is possible to model the dataset using multivariate Gaussians and subsequently apply the methods presented in this work to develop classifiers for such scenarios.

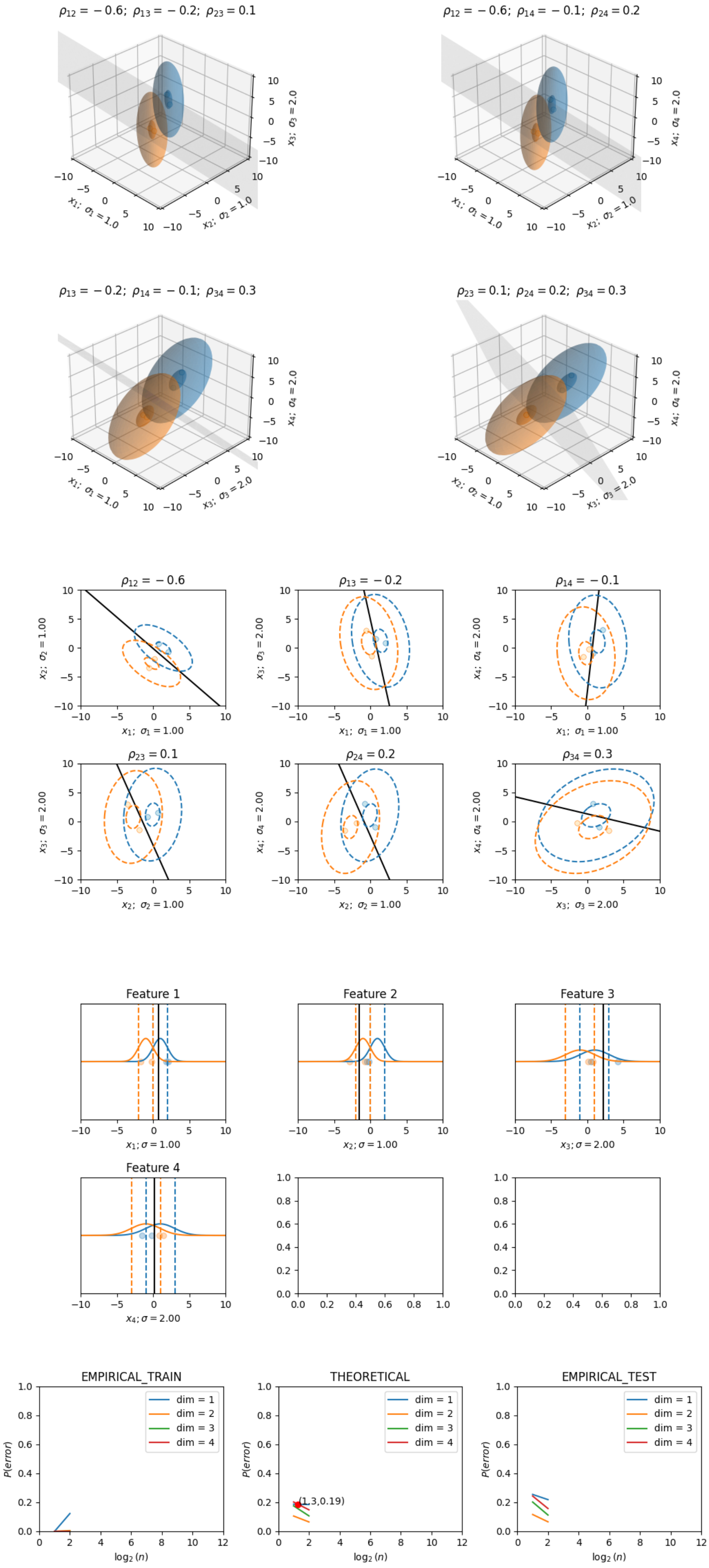


Figure 21 – Example $d > 3$; $\boldsymbol{\sigma} = [1, 1, 2, 2]$; $\boldsymbol{\rho} = [-0.3, -0.2, -0.1, 0.1, 0.2, 0.3]$; $n = 4$

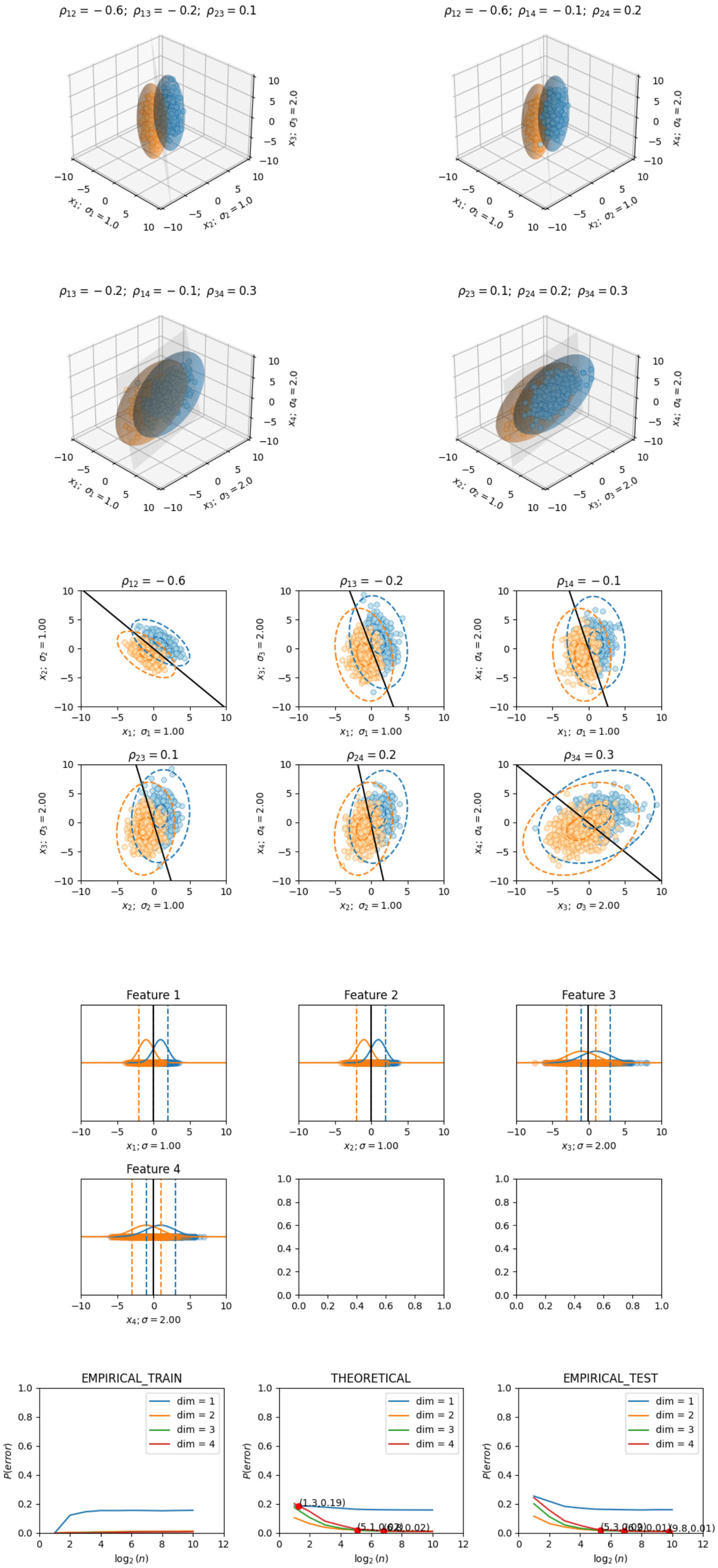


Figure 22 – Example $d>3$; $\boldsymbol{\sigma}=[1,1,2,2]$; $\boldsymbol{\rho}=[-0.3,-0.2,-0.1,0.1,0.2,0.3]$; $n=1024$

REFERENCES

BELKIN, M. et al. Reconciling modern machine-learning practice and the classical bias–variance trade-off. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 116, n. 32, p. 15849–15854, 2019.

BLANCHARD, G.; BOUSQUET, O.; MASSART, P. Statistical performance of support vector machines. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 36, n. 2, p. 489–531, 2008.

BOTTOU, L.; LIN, C.-J. Support vector machine solvers. **Large scale kernel machines**, Citeseer, v. 3, n. 1, p. 301–320, 2007.

BROWN, T. et al. Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020.

D'ASCOLI, S.; SAGUN, L.; BIROLI, G. Triple descent and the two kinds of overfitting: where & why do they appear? In: **NeurIPS**. [S.l.: s.n.], 2020.

DENG, Z.; KAMMOUN, A.; THRAMPOULIDIS, C. A model of double descent for high-dimensional binary linear classification. **Information and Inference: A Journal of the IMA**, 2021.

DEVROYE, L.; GYÖRFI, L.; LUGOSI, G. **A probabilistic theory of pattern recognition**. [S.l.]: Springer Science & Business Media, 2013. v. 31.

ENTEZARI-MALEKI, R.; REZAEI, A.; MINAEI-BIDGOLI, B. Comparison of classification methods based on the type of attributes and sample size. **J. Convergence Inf. Technol.**, Citeseer, v. 4, n. 3, p. 94–102, 2009.

FARAGÓ, A.; LUGOSI, G. Strong universal consistency of neural network classifiers. **IEEE Transactions on Information Theory**, IEEE, v. 39, n. 4, p. 1146–1151, 1993.

GLASMACHERS, T. Universal consistency of multi-class support vector classification. **Advances in Neural Information Processing Systems**, v. 23, p. 739–747, 2010.

HANCZAR, B.; DOUGHERTY, E. R. The reliability of estimated confidence intervals for classification error rates when only a single sample is available. **Pattern Recognition**, Elsevier, v. 46, n. 3, p. 1067–1077, 2013.

HE, F. et al. One-Shot Distributed Algorithm for PCA With RBF Kernels. **IEEE Signal Processing Letters**, IEEE, v. 28, p. 1465–1469, 2021.

HSIEH, C.-J. et al. A dual coordinate descent method for large-scale linear SVM. In: **Proceedings of the 25th international conference on Machine learning**. [S.l.: s.n.], 2008. p. 408–415.

HUA, J. et al. Optimal number of features as a function of sample size for various classification rules. **Bioinformatics**, Oxford University Press, v. 21, n. 8, p. 1509–1515, 2005.

HUGHES, G. On the mean accuracy of statistical pattern recognizers. **IEEE Transactions on Information Theory**, IEEE, v. 14, n. 1, p. 55–63, 1968.

LIST, N.; SIMON, H. U. SVM-optimization and steepest-descent line search. In: CITESEER. **Proceedings of the 22nd Annual Conference on Computational Learning Theory**. [S.l.], 2009.

LOOG, M.; VIERING, T.; MEY, A. Minimizers of the empirical risk and risk monotonicity. **Advances in Neural Information Processing Systems**, v. 32, p. 7478–7487, 2019.

MILLER, E. G.; MATSAKIS, N. E.; VIOLA, P. A. Learning from one example through shared densities on transforms. In: IEEE. **Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)**. [S.l.], 2000. v. 1, p. 464–471.

NAKKIRAN, P. et al. Optimal regularization can mitigate double descent. In: **International Conference on Learning Representations**. [S.l.: s.n.], 2020.

NOH, H.; RAJAGOPAL, R.; KIREMIDJIAN, A. Sequential structural damage diagnosis algorithm using a change point detection method. **Journal of Sound and Vibration**, Elsevier, v. 332, n. 24, p. 6419–6433, 2013.

PINHEIRO, J. I. D. Aprendendo a classificar com poucos atributos e amostras. **Tese de doutorado**, 2021. Universidade Federal do Rio de Janeiro.

R Documentation. 2021. <https://www.rdocumentation.org/packages/e1071>.

VERT, R.; VERT, J.-P.; SCHÖLKOPF, B. Consistency and convergence rates of one-class SVMs and related algorithms. **Journal of Machine Learning Research**, v. 7, n. 5, 2006.

WANG, Y. et al. Generalizing from a few examples: A survey on few-shot learning. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 53, n. 3, p. 1–34, 2020.

WILLETT, P.; SWASZEK, P. F.; BLUM, R. S. The good, bad and ugly: distributed detection of a known signal in dependent gaussian noise. **IEEE Transactions on signal processing**, IEEE, v. 48, n. 12, p. 3266–3279, 2000.

ZHANG, S. et al. Polygon structure-guided hyperspectral image classification with single sample for strong geometric characteristics scenes. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, 2021.

A PROBLEM CONTEXTUALIZATION AND FORMALIZATION

A.1 PROBLEM FORMULATION

The classification process involves two stages: training the classifier with labeled samples and then inferring the classes of unlabeled samples. Our two main variables of interest are the dataset cardinality, denoted by n, which corresponds to the number of samples in the training set, and the dataset dimensionality, denoted by d, which corresponds to the number of features in each sample. One of our objectives is to evaluate how n and d simultaneously affect the classifier's performance, as measured by its expected error probability.

A.1.1 Context

We consider a binary classification problem, where classes are denoted by labels -1 and $+1$. The dataset $\mathcal{D}$ contains n ordered pairs $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, for $i = 1, \dots, n$. The dataset $\mathcal{D}$ is used to train a classifier that minimizes the error probability. Let X and Y be the random variables corresponding to the feature vector and target class, respectively, assuming the joint probability distribution for the pair (X, Y) is unknown.

A classifier is a function $h : \boldsymbol{x} \to h(\boldsymbol{x})$ that assigns a label $h(\boldsymbol{x}) \in \{-1, +1\}$ to any feature vector $\boldsymbol{x} \in \mathbb{R}^d$. The classifier's error probability (or loss) is denoted by $L(h)$ and is given by $L(h) = P(Y \neq h(X))$. A dictionary H (also known as a hypothesis set or search space bias) is a set of classifiers (e.g., linear, quadratic, or rectangular classifiers). The learning problem is to choose a classifier from the dictionary H as the "solution" to our classification problem. It is well known that the Bayes classifier, which classifies each input according to $\text{argmax}_y P(Y = y|X = x)$, is globally optimal (DEVROYE; GYÖRFI; LUGOSI, 2013; FARAGÓ; LUGOSI, 1993); the Bayes error is the minimum error probability. We denote the Bayes classifier and its error probability by $h^{(B)}$ and $L(h^{(B)})$, or h^* and $L(h^*)$.

A.1.2 How to Evaluate the Efficiency of a Classifier

What should be our goal when facing a binary classification problem? Common sense suggests 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's assume that the available data is used to calibrate a classifier whose empirical error rate is as low as possible when applied to the dataset D. Then, of course, the best way to evaluate the efficiency of this classifier would be to predict its empirical error rate whenever applied to another independent dataset D'. If we are lucky enough to obtain a dataset with many observations, the usual solution would be to randomly split the available dataset into training and test sets. This would avoid underestimating the true error rate by using the same data to calibrate the classifier and also evaluate its effectiveness. In this context, it is important to distinguish between the empirical and theoretical error rates associated with a given classifier: the empirical rate is the training rate (which tends to underestimate the true error rate), while the test rate provides a more reliable estimate of its expected error rate. Furthermore, if it is not possible to obtain a dataset with many observations, a resampling technique (such as cross-validation, bootstrap, etc.) could be used to avoid such bias in estimating classification error rates. Now, what if we work with simulated data coming from a known probabilistic model? In this case, for a given classifier, it is always possible to use probability theory to calculate (eventually using numerical methods) its theoretical error rate without having to evaluate its effectiveness through test data. As this is a methodological work, we chose to work with simulated data, which greatly facilitates the task of evaluating the effectiveness of each classifier impartially. Fortunately, conclusions based on this type of approach are also applicable to concrete situations where we work with real data. On the other hand, when facing a binary classification problem, what are our "control knobs"? Among them, we can mention:

- The dataset cardinality n, i.e., the number of available observations.
- The problem dimensionality d, i.e., the number of features available in each observation.
- The discriminative power of each feature, alone or in the presence of other features.
- Dictionary H, from which we will choose our classifier.

It is worth asking how a particular choice of combination of these input parameters will affect the expected classification error rate in our mathematical model.

B BIVARIATE GAUSSIAN AND SVM CLASSIFIER

To evaluate the generalization power of the classifiers considered, we have at least three alternatives:

1. Sample D from a known probabilistic model, as in (NAKKIRAN et al., 2020; D'ASCOLI; SAGUN; BIROLI, 2020), and use the training sample D itself to evaluate the empirical classification performance of h, as in (C.1).
2. Evaluate the performance of h using equations from the probabilistic linear model like (D.1), (D.6), or (D.28).
3. Evaluate the performance of h on a test dataset D', generated from the same probabilistic model used to sample D.

B.1 BIVARIATE AND TRIVARIATE GAUSSIANS CONSIDERED IN THIS WORK

In particular, consider a two-dimensional classification problem ($d = 2$) or a three-dimensional problem ($d = 3$), in which samples from each class are drawn from a bivariate or trivariate Gaussian distribution. We assume equal priors for the two classes, i.e., $P(Y = +1) = P(Y = -1) = 0.5$.

B.1.1 Bivariate Case

The conditional distribution of the feature vector $X = (X_1, X_2)$ given Y is given by

$$(X_1, X_2) \sim \begin{cases} \mathcal{N}((+1,+1), \Sigma), & \text{if } Y = +1, \\ \mathcal{N}((-1,-1), \Sigma), & \text{if } Y = -1, \end{cases} \tag{B.1}$$

where $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a bivariate Gaussian distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^2$ and covariance matrix Σ,

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \tag{B.2}$$

with $\sigma_1 = 1$, $\sigma_2 \geq 1$ and $|\rho| \leq 1$. Note that σ_2 is the conditional standard deviation of feature X_2, given Y. If $\sigma_2 > 1$, the feature X_2 has less discriminative power than feature X_1: the larger the value of σ_2, the less informative feature X_2 is.

Note that we consider balanced datasets, with the same number of samples in each class.

B.1.2 Trivariate Case

In the three-dimensional case, we have as parameters $\sigma_1, \sigma_2, \sigma_3, \rho_{12}, \rho_{13}, \rho_{23}$ that constitute the covariance matrix,

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{bmatrix}.$$

Unless stated otherwise, we focus on the two-dimensional and three-dimensional cases in this work.

B.2 CLASSIFIER **SVM**

Our dictionary H consists of all linear classifiers, that is, straight lines. A naive approach to learning involves finding the line that minimizes the empirical error rate, for example, through a grid search. However, the problem may admit multiple solutions, requiring a well-founded strategy to break ties (BLANCHARD; BOUSQUET; MASSART, 2008). Therefore, we consider linear support vector machines (**SVM**), which have a series of desirable properties, including:

1. a well-founded strategy to determine the best separator, maximizing the distance, known as the *margin*, between the separating line and the two classes of points to be separated (BLANCHARD; BOUSQUET; MASSART, 2008);

2. polynomial computational cost (BOTTOU; LIN, 2007; LIST; SIMON, 2009), and

3. convergence to an optimal solution under mild conditions (HSIEH et al., 2008; GLASMACHERS, 2010; VERT; VERT; SCHÖLKOPF, 2006).

The linear **SVM** depends on a single parameter, the cost of incorrect classification, which appears as a regularization term in the Lagrange formulation and is denoted by γ (R..., 2021). Throughout this work, we keep $\gamma = 1$, its default value; we experimented with other values, but the results remained approximately the same.

C TYPES OF ERRORS

Next, we consider the types of errors, also called loss functions. In particular, we are interested in understanding how:

- the classifier's accuracy increases as the number of samples increases,
- the accuracy of error measurement improves as the number of samples increases.

Another aspect that should not be neglected is the precision with which we can estimate the error rate of our classification procedure. And, of course, it is also interesting to investigate how each input parameter affects this accuracy. This is what we will discuss now. In particular, we will investigate what happens to the accuracy of the error rate estimate as the cardinality of the dataset n tends to infinity.

Thus, from now on, let H be a specific dictionary. Furthermore, suppose we have a dataset D, which is representative of the phenomenon under study and consists of pairs (x_i, y_i), $i = 1, 2, ..., n$, where $x_i \in \mathbb{R}^d$ 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 $\hat{L}(h)$, that is, the proportion of misclassified observations in the dataset D:

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

Let $\hat{h}^{(D)}$ be the best empirical classifier in the dictionary H, that is, the classifier in H with the minimum empirical error rate. Under these conditions, we can define four error rates:

- $\min_{h \in H} L(h)$ = the lowest **Theoretical Error** rate achievable in H
- $L(\hat{h}^{(D)})$ = the theoretical error rate of the classifier $\hat{h}^{(D)}$ (eq. (D.7))
- $\hat{L}(\hat{h}^{(D)})$ = the empirical error rate of the classifier $\hat{h}^{(D)}$ (eq. 2.1)
- $\hat{L}(\hat{h}^{(D)}, D')$ = the error rate of the classifier $\hat{h}^{(D)}$ using a test dataset D' (eq. 2.2)

It is known that these losses necessarily obey an order, for any size of D:

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

From the results of our experiment, we observe that, for a very large dataset ($n \to \infty$):

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

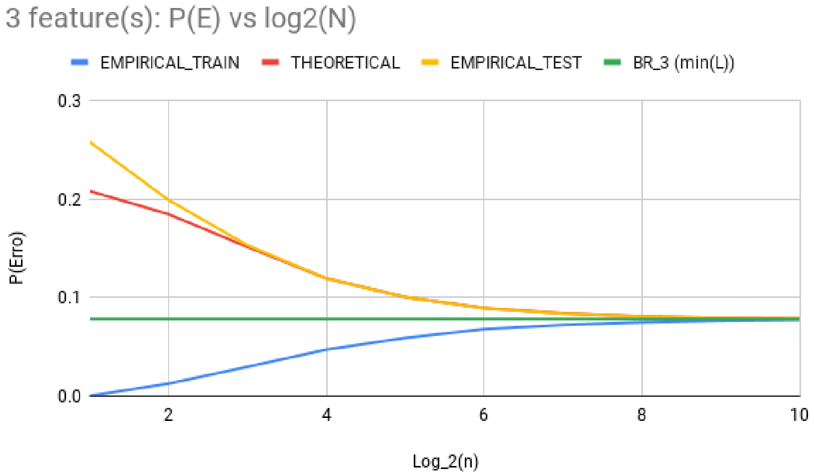


Figure 23 – Types of errors; $\boldsymbol{\sigma} = [1, 1, 2]$ and $\boldsymbol{\rho} = [0, 0.3, 0.3]$

For smaller datasets, the theoretical error rate is lower than the empirical error rate using a distinct test dataset D':

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

$\min_{h \in H} L(h)$ is also referred to in this work as the Bayes error rate (also known as Bayesian risk), which is the minimum theoretical loss generally achievable for this particular classification problem, detailed in appendix D.

In Figure 23, we have an example of a graph produced by simulating a case. This graph ensures the correctness of the simulator concerning the calculation of Bayes risk, performed analytically, as proposed by the studies of Professor João I. Pinheiro. We can observe that as the number of observations increases, the loss function curves approach Bayes risk.

D ANALYTICAL MODEL FOR THE MINIMUM ACHIEVABLE THEORETICAL CLASSIFICATION ERROR: SPECIAL CASES

Next, we consider the Bayes Risk of a classifier for a sample of infinite size ($n = \infty$). Specifically, we focus on deriving an expression for the minimum theoretical error achievable by a classifier in the 1D, 2D, and 3D cases.

D.1 1D: ONE ATTRIBUTE

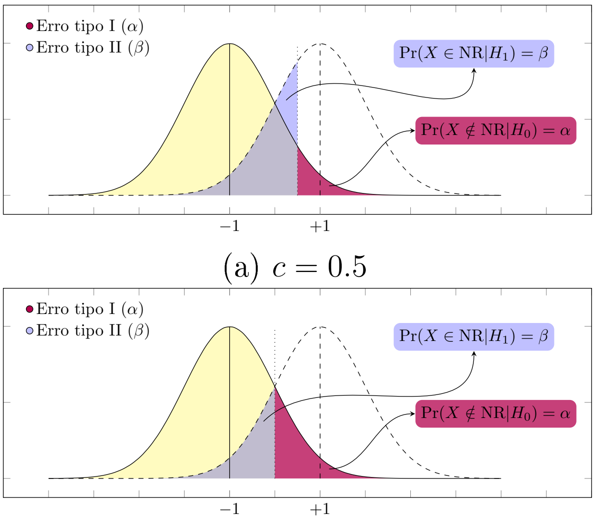
We begin by considering the case where only the attribute x_1 is available. We have a one-dimensional classification problem, in which we seek the best cutoff point c. The probability of error, in the case where the samples of the two classes are given by Gaussian distributions with variance 1 and means -1 and +1, is given by:

$$L_1(c) = \frac{1}{2}\Phi(c-1) + \frac{1}{2}\left(1 - \Phi(c+1)\right) = \frac{1}{2}(1 + \Phi(c-1) - \Phi(c+1)) \tag{D.1}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a Gaussian with mean 0 and variance 1. The optimal separator is $c^* = 0$, and

$$L_1(0) = 1 - \Phi(1) = 0.1586553. \tag{D.2}$$

Below, we present the derivation of (D.1). To do so, Figure 24 illustrates the concepts above using hypothesis testing terminology. The null hypothesis H_0 corresponds to the



(a) $c = 0.5$

(b) $c = 0$

Figure 24 – The non-rejection region of H_0 is NR, where α is the level of significance, i.e., the probability of incorrectly rejecting the null hypothesis H_0. $1 - \beta$ is the power of the test, i.e., the probability of correctly rejecting the null hypothesis H_0.

sample X belonging to class -1, i.e., $X \sim \mathcal{N}(-1, 1)$, while the alternative hypothesis H_1 corresponds to the sample X belonging to class +1, i.e., $X \sim \mathcal{N}(+1, 1)$. The non-rejection region of H_0 is denoted by NR. The probability of error is given by:

$$
\begin{aligned}
L_1(c) &= P(H_1)P(X \in NR|H_1) + P(H_0)P(X \notin NR|H_0) = && \text{(D.3)}\\
&= \frac{1}{2}\beta + \frac{1}{2}\alpha = && \text{(D.4)}\\
&= \frac{1}{2}\left(\Phi(c-1) + (1 - \Phi(c+1))\right). && \text{(D.5)}
\end{aligned}
$$

Indeed, observing Figure 24, we see that the blue region, to the left of c, with probability β, is such that:

$$P(X \in NR|H_1) = \beta = \Phi(c-1).$$

Similarly, we see that the red region, to the right of c, with probability α, is such that:

$$P(X \notin NR|H_0) = \alpha = 1 - \Phi(c+1).$$

D.2 2D: TWO ATTRIBUTES

D.2.1 Error Expression

Next, we consider the case where two attributes are available. Given a classifier characterized by the line $x_2 = a + bx_1$:

$$L_2(a, b) = \frac{1}{2}\left(1 - \Phi\left(\frac{|-a-b+1|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|a-b+1|}{\sqrt{\Delta}}\right)\right). \qquad \text{(D.6)}$$

The above error expression considers two classes, with means $\mu_+ = (+1, +1)$ and $\mu_- = (-1, -1)$.

D.2.2 Relation Between (D.6) and Section 2.4.2

Next, we relate the error expression (D.6) with the terminology from Section 2.4.2. We have:

$$bx_1 - x_2 + a = 0$$

or equivalently,

$$w_1x_1 + w_2x_2 - \tilde{b} = 0.$$

Thus,

$$\tilde{b} = -a, \quad \kappa = -a$$

and

$$w_1 = -b/(-1) = b, \quad w_2 = -1.$$

The distance calculations are as follows:

$$\text{Distance}(+) = \frac{\left|w_1 - 1 - \tilde{b}\right|}{\sqrt{\Delta}} = \frac{|b-1+a|}{\sqrt{\Delta}}, \quad \text{Distance}(-) = \frac{\left|w_1 - 1 + \tilde{b}\right|}{\sqrt{\Delta}} = \frac{|b-1-a|}{\sqrt{\Delta}}.$$

Thus, we can express the **Theoretical Error** as a function of the weight vector **w** and the bias b:

$$L_d(\mathbf{w}, b) = \frac{1}{2}\left(1 - \Phi\left(\frac{|\sum_{i=1}^{i=d} w_i - \tilde{b}|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|\sum_{i=1}^{i=d} w_i + \tilde{b}|}{\sqrt{\Delta}}\right)\right) \tag{D.7}$$

Note that

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} = \boldsymbol{\lambda}\boldsymbol{\lambda}^T.$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \sigma_1 & 0 \\ \frac{\rho\sigma_1\sigma_2}{\sigma_1} & \sqrt{\sigma_2^2 - (\rho\sigma_1\sigma_2/\sigma_1)^2} \end{bmatrix}, \quad \boldsymbol{\lambda}^T = \begin{bmatrix} \sigma_1 & \frac{\rho\sigma_1\sigma_2}{\sigma_1} \\ 0 & \sqrt{\sigma_2^2 - (\rho\sigma_1\sigma_2/\sigma_1)^2} \end{bmatrix}.$$

$$\tilde{\boldsymbol{\delta}} = \boldsymbol{\lambda}^T \cdot \begin{bmatrix} b \\ -1 \end{bmatrix} = \begin{bmatrix} \tilde{\delta}_1 \\ \tilde{\delta}_2 \end{bmatrix}.$$

Thus,

$$\Delta = \tilde{\delta}_1^2 + \tilde{\delta}_2^2$$

or equivalently,

$$\Delta = (-b\sigma_1 + \rho\sigma_2)^2 + \sigma_2^2 - (\rho\sigma_2)^2. \tag{D.8}$$

D.2.2.1 Probability of Gaussian Generating a Point Above a Given Line

Next, we seek $P(X_2 > bX_1 + a)$ where $(X_1, X_2) \sim N(\mu_1, \mu_2, \Sigma)$.

Teorema 1. *If* $(X_1, X_2) \sim N(\mu_1, \mu_2, \Sigma)$*, then*

$$P(X_2 > bX_1 + a) = P(Z_2 > z_2),$$

where $Z_2 \sim N(0, 1)$ *and*

$$z_2 = \frac{b\mu_1 + a - \mu_2}{\sqrt{(\sigma_2\rho - b\sigma_1)^2 + \sigma_2^2(1-\rho^2)}} = \frac{b\mu_1 + a - \mu_2}{\sqrt{\Delta}}.$$

The proof proceeds in four steps:
1. **First Step:** Let $(X_1, X_2) \sim N(\mu_1, \mu_2, \Sigma)$ where

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$

Then,

$$X_1 = \mu_1 + \sigma_1 X_1', \quad X_2 = \mu_2 + \sigma_2 X_2'$$

where $X_1', X_2' \sim N(0,1)$ with correlation coefficient ρ.
2. **Second Step:**

$$X_2' = \rho X_1' + \sqrt{1-\rho^2} Z$$

where $Z \sim N(0,1)$ is independent of X_1'.
3. **Third Step:** Write the event $X_2 > bX_1 + a$ in the form:

$$X_1' \geq \beta Z + \alpha.$$

In detail,

$$X_2 > bX_1 + a \tag{D.9}$$

$$\mu_2 + \sigma_2\left(\rho X_1' + \sqrt{1-\rho^2}Z\right) > b(\mu_1 + \sigma_1 X_1') + a \tag{D.10}$$

$$\sigma_2\rho X_1' > b(\mu_1 + \sigma_1 X_1') + a - \mu_2 - \sigma_2\sqrt{1-\rho^2}Z \tag{D.11}$$

$$(\sigma_2\rho - b\sigma_1)X_1' > b\mu_1 + a - \mu_2 - \sigma_2\sqrt{1-\rho^2}Z \tag{D.12}$$

Thus, if $\sigma_2\rho - b\sigma_1 > 0$:

$$\alpha = \frac{b\mu_1 + a - \mu_2}{\sigma_2\rho - b\sigma_1}, \quad \beta = \frac{-\sigma_2\sqrt{1-\rho^2}}{\sigma_2\rho - b\sigma_1}.$$

4. **Fourth Step:**

$$Z_1 = \frac{X_1' - \beta Z}{\sqrt{1+\beta^2}}, \quad z_1 = \frac{\alpha}{\sqrt{1+\beta^2}}.$$

Note that $Z_1 \sim N(0,1)$ and:

$$P(X_2 > bX_1 + a) = P(Z_1 > z_1).$$

The reasoning above holds if $\sigma_2\rho - b\sigma_1 > 0$. In this case:

$$P(X_2 > bX_1 + a) = P(Z_1 > z_1)$$

where:

$$z_1 = \frac{\alpha}{\sqrt{1+\beta^2}},$$

and:

$$\alpha = \frac{b\mu_1 + a - \mu_2}{\sigma_2\rho - b\sigma_1}, \quad \beta = \frac{-\sigma_2\sqrt{1-\rho^2}}{\sigma_2\rho - b\sigma_1}.$$

If $\sigma_2\rho - b\sigma_1 < 0$, then:

$$P(X_2 > bX_1 + a) = P(Z_1 < z_1),$$

and the final result still holds because:

$$P(Z_1 < z_1) = P(Z_1 > -z_1).$$

In this case, where $\sigma_2\rho - b\sigma_1 < 0$, we have:

$$z_2 = -z_1 = \frac{b\mu_1 + a - \mu_2}{\sqrt{(\sigma_2\rho - b\sigma_1)^2 + \sigma_2^2(1-\rho^2)}}.$$

An intuitive way to understand the proof above is to visualize the original Gaussian, the normalized Gaussian, and the normalized Gaussian with the error region rotated. This visualization is presented in Figure 25.

D.2.2.2 Special Case 1: Proof of (D.6)

Recall that in this work, we consider two classes. In class 1, we have $\mu_1 = \mu_2 = 1$, and in class 2, we have $\mu_1 = \mu_2 = -1$. Thus, $-b + a + 1$ and $b + a - 1$ have opposite signs.

Case a) $b \leq 1$, i.e., $b + a - 1 \leq 0 \leq -b + a + 1$.

In this case, applying Theorem 1 for class 2:

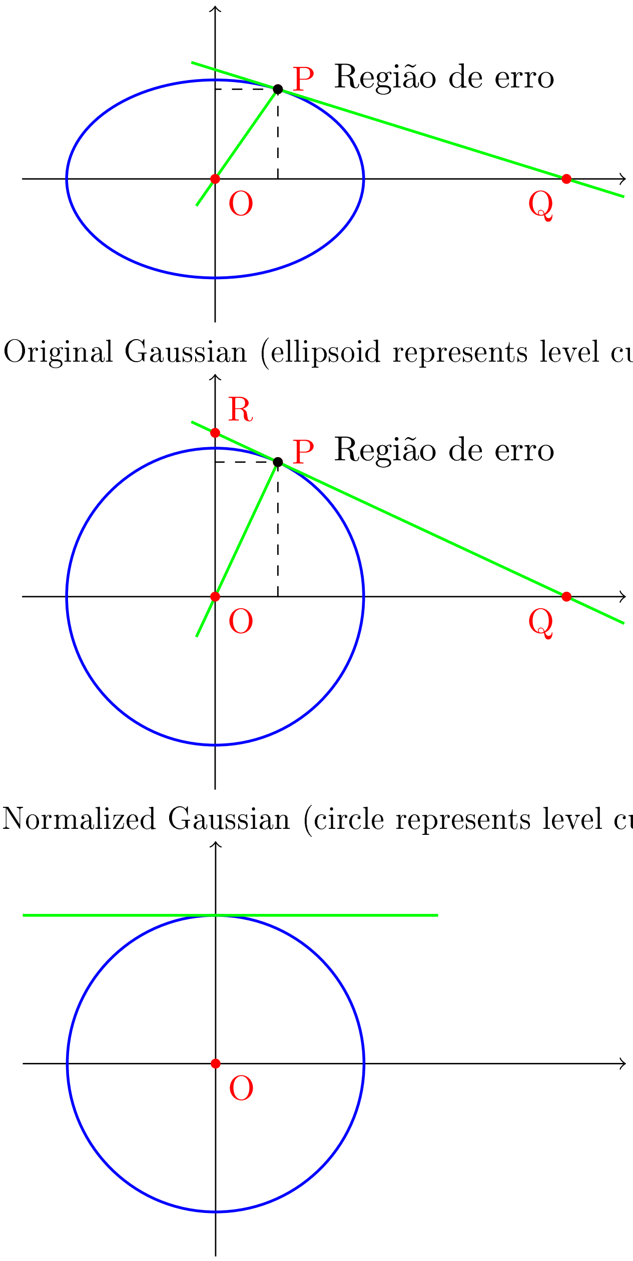
$$z_2 = \frac{b\mu_1 + a - \mu_2}{\sqrt{\Delta}} = \frac{-b + a + 1}{\sqrt{\Delta}} = \frac{|-b + a + 1|}{\sqrt{\Delta}}.$$

Thus:

$$P(\text{error}|-) = P(Z_2 > z_2) = 1 - P(Z_2 < z_2) = 1 - P\left(Z_2 < \frac{|-b+a+1|}{\sqrt{\Delta}}\right)$$

as expected (rightmost term in (D.6)).

Next, for class 1, by Theorem 1:



(a) Original Gaussian (ellipsoid represents level curve)

(b) Normalized Gaussian (circle represents level curve)

(c) Normalized Gaussian with rotated error region

Figure 25 – Finding the **Theoretical Error** for a bidimensional linear classifier

$$z_2 = \frac{b\mu_1 + a - \mu_2}{\sqrt{\Delta}} = \frac{b + a - 1}{\sqrt{\Delta}} = -\frac{|b + a - 1|}{\sqrt{\Delta}} = -\frac{|-b - a + 1|}{\sqrt{\Delta}}.$$

The rest of the derivation follows similarly:

$$P(\text{error}|+) = 1 - P(Z_2 > z_2) = 1 - P(Z_2 < -z_2) = 1 - P\left(Z_2 < \frac{|-b - a + 1|}{\sqrt{\Delta}}\right)$$

as expected (leftmost term in (D.6)).

In conclusion:

$$L_2(a,b) = \frac{1}{2}P\left(X_2^{(+)} < bX_1^{(+)} + a\right) + \frac{1}{2}P\left(X_2^{(-)} > bX_1^{(-)} + a\right) \tag{D.13}$$

$$= \frac{1}{2}\left(1 - P\left(X_2^{(+)} > bX_1^{(+)} + a\right)\right) + \frac{1}{2}P\left(X_2^{(-)} > bX_1^{(-)} + a\right) \tag{D.14}$$

$$= \frac{1}{2}\left(1 - \Phi\left(\frac{|-b-a+1|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|a-b+1|}{\sqrt{\Delta}}\right)\right) \tag{D.15}$$

$$= \frac{1}{2}\left(1 - \Phi\left(\frac{-a-b+1}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{a-b+1}{\sqrt{\Delta}}\right)\right) \tag{D.16}$$

The above expression matches the one we aimed to derive, i.e., (D.6).

Case b) $b \geq 1$, i.e., $b + a - 1 \geq 0 \geq -b + a + 1$.

For class 2, by Theorem 1:

$$z_2 = \frac{b\mu_1 + a - \mu_2}{\sqrt{\Delta}} = \frac{-b + a + 1}{\sqrt{\Delta}} = -\frac{|-b+a+1|}{\sqrt{\Delta}}.$$

Thus:

$$P(\text{error}|-) = 1 - P(Z_2 > z_2) = 1 - P(Z_2 < -z_2) = 1 - P\left(Z_2 < \frac{|-b+a+1|}{\sqrt{\Delta}}\right)$$

as expected (leftmost term in (D.6)).

Next, for class 1, by Theorem 1:

$$z_2 = \frac{b\mu_1 + a - \mu_2}{\sqrt{\Delta}} = \frac{b + a - 1}{\sqrt{\Delta}} = \frac{|b+a-1|}{\sqrt{\Delta}} = \frac{|-b-a+1|}{\sqrt{\Delta}}.$$

The rest of the derivation follows similarly:

$$P(\text{error}|+) = P(Z_2 > z_2) = 1 - P(Z_2 < z_2) = 1 - P\left(Z_2 < \frac{|-b-a+1|}{\sqrt{\Delta}}\right)$$

as expected (rightmost term in (D.6)).

In conclusion:

$$L_2(a,b) = \frac{1}{2}P\left(X_2^{(-)} < bX_1^{(-)} + a\right) + \frac{1}{2}P\left(X_2^{(+)} > bX_1^{(+)} + a\right) \tag{D.17}$$

$$= \frac{1}{2}\left(1 - P\left(X_2^{(-)} > bX_1^{(-)} + a\right)\right) + \frac{1}{2}P\left(X_2^{(+)} > bX_1^{(+)} + a\right) \tag{D.18}$$

$$= \frac{1}{2}\left(1 - \Phi\left(\frac{|-b+a+1|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|-a-b+1|}{\sqrt{\Delta}}\right)\right) \tag{D.19}$$

$$= \frac{1}{2}\left(1 - \Phi\left(\frac{-a+b-1}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{a+b-1}{\sqrt{\Delta}}\right)\right) \tag{D.20}$$

The above expression matches the one we aimed to derive, i.e., (D.6).

D.2.2.3 Special Case 2

In the particular case where $\sigma_1 = 1$, $\sigma_2 = \delta$, $\rho_{12} = \rho$, we have:

$$\Sigma = \begin{bmatrix} 1 & \rho\delta \\ \rho\delta & \delta^2 \end{bmatrix} = \boldsymbol{\lambda}\boldsymbol{\lambda}^T.$$

$$\boldsymbol{\lambda} = \begin{bmatrix} 1 & 0 \\ \rho\delta & \delta\sqrt{(1-\rho^2)} \end{bmatrix}$$

$$\tilde{\boldsymbol{\delta}} = \boldsymbol{\lambda}^T \cdot \begin{bmatrix} b \\ -1 \end{bmatrix} = \begin{bmatrix} \tilde{\delta}_1 \\ \tilde{\delta}_2 \end{bmatrix}$$

Thus:

$$\Delta = \tilde{\delta}_1^2 + \tilde{\delta}_2^2$$

or equivalently:

$$\Delta = (b - \rho\delta)^2 + \delta^2(1-\rho^2)$$

In this particular case, with $\sigma_1 = 1$, $\sigma_2 = \delta$, $\rho_{12} = \rho$, it can be verified that the optimal solution (a^*, b^*) that minimizes the error (D.6) is given by a line through the origin, $a^* = 0$, with a slope of $b^* = \delta(\delta - \rho)/(\delta\rho - 1)$.

We can also derive an analytical expression for the Bayes error:

$$\frac{|0 - b^* + 1|}{\sqrt{\Delta}} = \frac{b^* - 1}{\sqrt{S}} = \frac{\delta(\delta-\rho)/(\delta\rho-1) - 1}{\sqrt{(\rho\delta - \delta(\delta-\rho)/(\delta\rho-1))^2 + \delta^2(1-\rho^2)}} \tag{D.21}$$

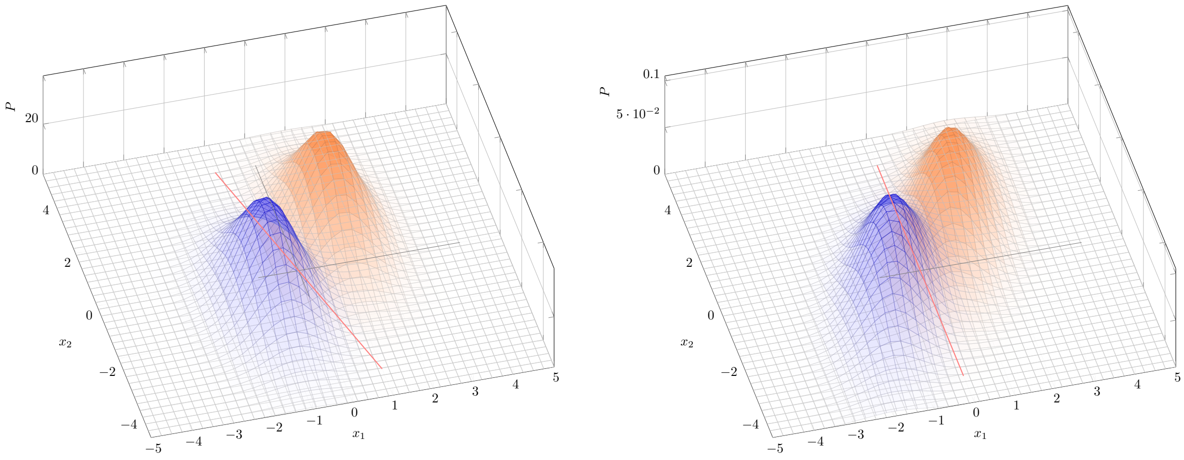
$$= \frac{1}{\delta}\sqrt{\frac{\delta^2 - 2\rho\delta + 1}{1-\rho^2}}. \tag{D.22}$$

Thus:

$$L_2\left(0, \frac{\delta(\delta-\rho)}{\delta\rho - 1}\right) = 1 - \Phi\left(\frac{1}{\delta}\sqrt{\frac{\delta^2 - 2\rho\delta + 1}{1-\rho^2}}\right). \tag{D.23}$$

D.2.3 Illustrating Results

Two scenarios with $\delta = 2$ are illustrated in Figure 26. We can see that as ρ increases between 0 and 0.5, the relevance of the second attribute decreases and the problem becomes more challenging. In particular, when $\rho = 0.5$, the second attribute is irrelevant for classification purposes, and it is clearly advantageous to collect more observations rather than an additional attribute for training a classifier.



(a) $\rho = 0.01$

(b) $\rho = 0.5$

Figure 26 – Gaussians separated by a line, where $\delta = 2$ and (a) $\rho \approx 0$ and (b) $\rho = 0.5$

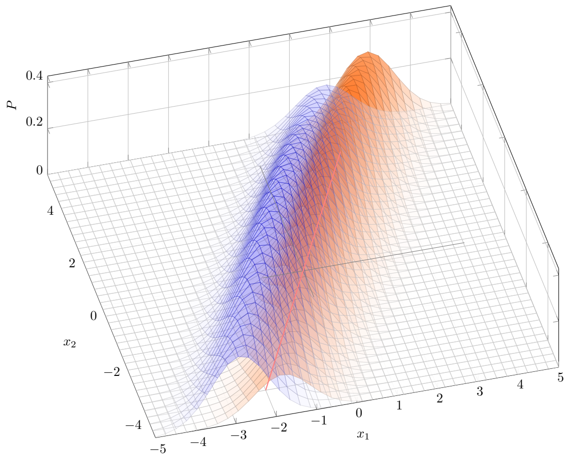
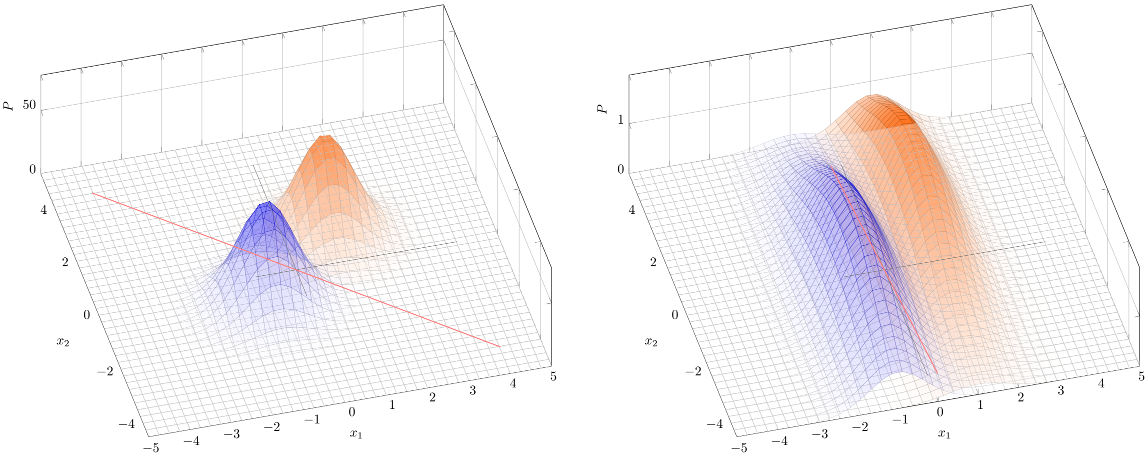


Figure 27 – Gaussians separated by a line when $\rho = 0.95$

In Figure 27, still with $\delta = 2$, we consider the case $\rho = 0.95$. We note that increasing ρ causes the second attribute to become relevant again. In the extreme case where $\rho = 1$, we obtain a local minimum for the classification error. Intuitively, this occurs because the utility of the second attribute for classification purposes increases as ρ grows from 0.5 to 1. With a useful second attribute to aid in classifier construction, the classification error decreases.

Figure 28 illustrates the impact of δ on the utility of X_2. The greater the value of δ, the lower the utility of X_2. Intuitively, when the attribute X_2 has high dispersion, it does not significantly contribute to increasing the power of the classifier. In this case, collecting additional samples is more useful for improving the classifier than adding the attribute X_2.



(a) $\delta = 1$

(b) $\delta = 4$

Figure 28 – Impact of δ on the utility of X_2, assuming $\rho = 0.01$

D.3 3D: THREE ATTRIBUTES

D.3.1 General Solution

As in previous sections, we assume that the conditional covariance matrices of the two considered classes are equal. The 3D classification problem can be formulated as follows:

$$(X_1, X_2, X_3) \sim \begin{cases} \mathcal{N}((\mu_1^+, \mu_2^+, \mu_3^+), \Sigma), & \text{if } Y = +1 \\ \mathcal{N}((\mu_1^-, \mu_2^-, \mu_3^-), \Sigma), & \text{if } Y = -1 \end{cases} \tag{D.24}$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{bmatrix} \tag{D.25}$$

We will also assume, as in previous sections:

$$P(Y = +1) = P(Y = -1) = \frac{1}{2} \tag{D.26}$$

and

$$\mu^+ = (1, 1, 1), \quad \mu^- = (-1, -1, -1) \tag{D.27}$$

Given a 3D classifier, defined by the plane: $X_3 = d + eX_1 + fX_2$, the corresponding expected error probability is calculated by:

$$\begin{aligned} L_3(d, e, f) &= \frac{1}{2}(1 - \Phi(\text{Distance}(+))) + \frac{1}{2}(1 - \Phi(\text{Distance}(-))) = \\ &= \frac{1}{2}\left(1 - \Phi\left(\frac{|d + e + f - 1|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|-d + e + f - 1|}{\sqrt{\Delta}}\right)\right) \end{aligned} \tag{D.28}$$

where:

- $\Phi(\cdot)$ is the cumulative distribution function of a Gaussian with zero mean and unit variance.
- Distance(+) and Distance(-) are the distances of the separating hyperplane h to the mean of the positive and negative classes, respectively.

The term Δ that appears in the denominator of the above expression is obtained through the Cholesky decomposition of the matrix Σ.

Teorema 2. *A covariance matrix Σ of dimension $M \times M$ can be decomposed, via Cholesky decomposition, into a lower triangular matrix $\boldsymbol{\lambda}$ such that $\boldsymbol{\lambda}\boldsymbol{\lambda}^T = \Sigma$.*

Thus, Δ is given by:

$$\Delta = A^2 + B^2 + \lambda_{33}^2 \tag{D.29}$$

where:

- $A = e\lambda_{11} + f\lambda_{21} - \lambda_{31}$
- $B = f\lambda_{22} - \lambda_{32}$
- $\lambda_{11} = \sigma_1$, $\lambda_{12} = 0$, $\lambda_{13} = 0$
- $\lambda_{21} = \rho_{12}\sigma_2$, $\lambda_{22} = \sigma_2\sqrt{1-\rho_{12}^2}$, $\lambda_{23} = 0$
- $\lambda_{31} = \rho_{13}\sigma_3$, $\lambda_{32} = \frac{(\rho_{23}-\rho_{12}\rho_{13})\sigma_3}{\sqrt{1-\rho_{12}^2}}$, $\lambda_{33} = \sigma_3\sqrt{1-\rho_{13}^2 - \frac{(\rho_{23}-\rho_{12}\rho_{13})^2}{(1-\rho_{12})^2}}$
- λ_{ij} are elements of $\boldsymbol{\lambda}$
- $\boldsymbol{\lambda}$: is the lower-triangular Cholesky factor of Σ such that $\Sigma = \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^T$
- $\tilde{\boldsymbol{\delta}} = \boldsymbol{\lambda}^T \cdot \begin{bmatrix} e \\ f \\ -1 \end{bmatrix} = \begin{bmatrix} e\lambda_{11} + f\lambda_{21} - \lambda_{31} \\ f\lambda_{22} - \lambda_{32} \\ -\lambda_{33} \end{bmatrix}$
- $\Delta = \sum_{\tilde{\delta}_i \in \tilde{\boldsymbol{\delta}}} \tilde{\delta}_i^2 = \tilde{\delta}_1^2 + \tilde{\delta}_2^2 + \tilde{\delta}_3^2 = (e\lambda_{11} + f\lambda_{21} - \lambda_{31})^2 + (f\lambda_{22} - \lambda_{32})^2 + (-\lambda_{33})^2$

Note that this formulation includes many (six) parameters, and the mathematical optimization problem becomes complex. In Appendix F, we consider four special scenarios in which we can simplify the Bayes error expression and present it in closed form. In general, the complexity of the above problem motivates the use of simulation, as considered in this work, to evaluate the impact of different parameters on the quality of the obtained classifier.

D.3.2 Optimal Solution

Now, suppose that the equation of the optimal separating plane is:

$$X_3 = d^* + e^* X_1 + f^* X_2 \tag{D.30}$$

To minimize $P(\text{Error})$ with respect to d^*, e^*, and f^*, we need to set their partial derivatives to zero. Then, it can be shown that:

- $d^* = 0$ (due to the general symmetry around the origin $(0, 0, 0)$)
- Both e^* and f^* depend on the six parameters that define the matrix Σ, namely: $\sigma_1, \sigma_2, \sigma_3, \rho_{12}, \rho_{13}, \rho_{23}$. Unlike the 1D and 2D cases, in the 3D case we were unable to obtain a closed-form mathematical expression for e^* and f^* (see Appendix F for special cases where closed-form solutions were possible).

In the particular case where $d = 0$, we obtain from (D.28):

$$L_3(0, e, f) = 1 - \Phi\left(\frac{|1 - e - f|}{\sqrt{\Delta}}\right) \tag{D.31}$$

Teorema 3. *The minimum probability of* ***Theoretical Error*** *(Bayes Risk) in the 3D case (trivariate Gaussian) is*

$$P(\mathit{Error}) = 1 - \Phi\left(\frac{|1 - e^* - f^*|}{\sqrt{\Delta}}\right)$$

The denominator Δ is characterized as a function of the Cholesky decomposition of the matrix Σ, as given in (D.29).

D.4 GENERAL CASE WITH d DIMENSIONS

Teorema 4. *If $(X_1, X_2, \ldots, X_d) \sim N(\mu_1, \mu_2, \ldots, \mu_d, \Sigma)$, then*

$$P\left(\tilde{b} > \sum_{i=1}^{d} X_i w_i\right) = P(Z_2 > z_2)$$

where $Z_2 \sim N(0, 1)$ and

$$z_2 = \frac{-\tilde{b} + \sum w_i \mu_i}{\sqrt{\Delta}}.$$

In the above expression, we have:

$$\Delta = \sum \tilde{\delta}_i^2$$

and

$$\Sigma = \boldsymbol{\lambda}\boldsymbol{\lambda}^T, \quad \tilde{\boldsymbol{\delta}} = \boldsymbol{\lambda}^T \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} \tilde{\delta}_1 \\ \tilde{\delta}_2 \\ \vdots \\ \tilde{\delta}_d \end{bmatrix}$$

Prova: The theorem above is a consequence of basic properties of affine transformations of multivariate Gaussians.[1] Indeed,

$$\sum X_i w_i \sim \mathcal{N}\left(\boldsymbol{w}^T\boldsymbol{\mu}, \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}\right) \tag{D.32}$$

Thus,

$$\frac{\sum X_i w_i - \boldsymbol{w}^T\boldsymbol{\mu}}{\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}}} \sim \mathcal{N}(0,1) \tag{D.33}$$

Therefore,

$$P\left(\tilde{b} > \sum_{i=1}^{d} X_i w_i\right) = P\left(\frac{\tilde{b} - \boldsymbol{w}^T\boldsymbol{\mu}}{\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}}} > \frac{\sum_{i=1}^{d} X_i w_i - \boldsymbol{w}^T\boldsymbol{\mu}}{\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}}}\right) = P(Z_2 < -z_2) = P(Z_2 > z_2) \tag{D.34}$$

Note also that

$$\sqrt{\boldsymbol{w}^T\Sigma\boldsymbol{w}} = \sqrt{\boldsymbol{w}^T\boldsymbol{\lambda}\boldsymbol{\lambda}^T\boldsymbol{w}} = \sqrt{\Delta}. \tag{D.35}$$

□

Recall that in this work we consider two classes. In class 1, we have $\mu_1 = \mu_2 = \ldots = \mu_d = 1$, and in class 2, we have $\mu_1 = \mu_2 = \ldots = \mu_d = -1$. Thus, $-\sum w_i + \tilde{b}$ and $\sum w_i + \tilde{b}$ have opposite signs. Given the above assumptions and the fact that both classes share the same covariance matrix Σ, we have the following theorem.

Teorema 5. *Given two classes of points generated by multivariate Gaussians, with means* $(+1, +1, \ldots, +1)$ *and* $(-1, -1, \ldots, -1)$ *and covariance matrix* Σ*, the theoretical error of the classifier characterized by the separating hyperplane* $\sum_{i=1}^{d} w_i X_i = \tilde{b}$ *is given by*

$$P(error) = \frac{1}{2}\left(1 - \Phi\left(\frac{|\tilde{b} - \sum_{i=1}^{d} w_i|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|-\tilde{b} - \sum_{i=1}^{d} w_i|}{\sqrt{\Delta}}\right)\right) \tag{D.36}$$

where $w_d = -1$.

Case a) $\sum w_i \leq 0$. Thus, since $\sum w_i + \tilde{b}$ and $-\sum w_i + \tilde{b}$ have opposite signs, $\sum w_i + \tilde{b} \leq 0 \leq -\sum w_i + \tilde{b}$.

In this case, the classification rule is as follows. The estimated class, $\hat{Y}$, is given by:

$$\hat{Y} = \begin{cases} +1, & \text{if } \sum X_i w_i < \tilde{b} \\ -1, & \text{otherwise.} \end{cases} \tag{D.37}$$

In particular, in the two-dimensional case, we have $X_2 = a + bX_1$ as the separating line, $w_2 = -1$, $w_1 = b$, $\tilde{b} = -a$, so we classify as $+1$ if $-X_2 + bX_1 < -a$ and as -1 otherwise.

Then, by Theorem 4, for class 2 we have:

$$z_2 = \frac{-\tilde{b} + \sum w_i \mu_i}{\sqrt{\Delta}} = \frac{-\tilde{b} - \sum w_i}{\sqrt{\Delta}} = \frac{|-\tilde{b} - \sum w_i|}{\sqrt{\Delta}}.$$

[1] <https://en.wikipedia.org/wiki/Multivariate_normal_distribution>

We conclude that:

$$P(\text{error}|-) = P(Z_2 > z_2) = 1 - P(Z_2 < z_2) = 1 - P\left(Z_2 < \frac{|-\tilde{b} - \sum w_i|}{\sqrt{\Delta}}\right)$$

as expected (the term on the right of (D.36)).

Now consider class 1. By Theorem 4,

$$z_2 = \frac{-\tilde{b} + \sum w_i \mu_i}{\sqrt{\Delta}} = \frac{-\tilde{b} + \sum w_i}{\sqrt{\Delta}} = -\frac{|-\tilde{b} + \sum w_i|}{\sqrt{\Delta}} = -\frac{|\tilde{b} - \sum w_i|}{\sqrt{\Delta}}.$$

The rest of the derivation follows similarly to conclude that:

$$P(\text{error}|+) = 1 - P(Z_2 > z_2) = 1 - P(Z_2 < -z_2) = 1 - P\left(Z_2 < \frac{|\tilde{b} - \sum w_i|}{\sqrt{\Delta}}\right)$$

as expected (the term on the left of (D.36)).

Concluding:

$$L_d(\boldsymbol{w}) = \frac{1}{2}P\left(X_d^{(+)} < \sum_{i=1}^{d-1} w_i X_i^{(+)} - \tilde{b}\right) + \frac{1}{2}P\left(X_d^{(-)} > \sum_{i=1}^{d-1} w_i X_i^{(-)} - \tilde{b}\right) \tag{D.38}$$

$$= \frac{1}{2}\left(1 - P\left(X_d^{(+)} < \sum_{i=1}^{d-1} w_i X_i^{(+)} - \tilde{b}\right)\right) + \frac{1}{2}P\left(X_d^{(-)} > \sum_{i=1}^{d-1} w_i X_i^{(-)} - \tilde{b}\right) \tag{D.39}$$

$$= \frac{1}{2}\left(1 - \Phi\left(\frac{|\tilde{b} - \sum w_i|}{\sqrt{\Delta}}\right)\right) + \frac{1}{2}\left(1 - \Phi\left(\frac{|-\tilde{b} - \sum w_i|}{\sqrt{\Delta}}\right)\right) \tag{D.40}$$

The above expression is equivalent to the one we want to obtain (D.36).

Case b) $\sum w_i \geq 0$. Thus, $\sum w_i + \tilde{b} \geq 0$.

In this case, the classification rule is as follows. The estimated class, $\hat{Y}$, is given by:

$$\hat{Y} = \begin{cases} -1, & \text{if } \sum X_i w_i < \tilde{b} \\ +1, & \text{otherwise.} \end{cases} \tag{D.41}$$

This case is analogous to the previous one, and we omit it for brevity.

E ADDITIONAL COMMENTS ON THE BAYES CLASSIFIER FOR THE CASE WHERE SAMPLES ARE DRAWN FROM BIVARIATE GAUSSIANS

Next, we consider the case where the samples are drawn from bivariate Gaussians, and the conditions of Section D.2.2.3 are satisfied.

We have $\sigma_1 = 1$, $\sigma_2 = \delta$, $\rho_{12} = \rho$. We also have:

$$\Sigma = \begin{bmatrix} 1 & \rho\delta \\ \rho\delta & \delta^2 \end{bmatrix} = \boldsymbol{\lambda}\boldsymbol{\lambda}^T$$

$$\boldsymbol{\lambda} = \begin{bmatrix} 1 & 0 \\ \rho\delta & \delta\sqrt{(1-\rho^2)} \end{bmatrix}$$

In addition,

$$\Delta = (b - \rho\delta)^2 + \delta^2(1-\rho^2).$$

Since in this case the optimal classifier $h^{(B)}$ belongs to the dictionary formed by lines in $\mathbb{R}^2$, the Bayes error, corresponding to the optimal classifier, is given by:

$$L(h^{(B)}) = L(h_2^*) = L_2\left(0, \frac{\delta(\delta-\rho)}{\delta\rho - 1}\right) = 1 - \Phi\left(\frac{1}{\delta}\sqrt{\frac{\delta^2 - 2\rho\delta + 1}{1-\rho^2}}\right). \tag{E.1}$$

By taking the partial derivative of the above expression with respect to ρ and setting the result to zero, we can determine the value of ρ that maximizes the loss. In fact, we conclude that the upper limit of $L(h^{(B)})$ is reached at $\rho = 1/\delta$, as detailed below.

Taking the derivative with respect to ρ of the term inside the square root in the above expression and setting the result to zero, we obtain:

$$\frac{d}{d\rho}\frac{\delta^2 - 2\rho\delta + 1}{1-\rho^2} = \frac{-2\delta}{1-\rho^2} - 2\rho\frac{\delta^2 - 2\rho\delta + 1}{(1-\rho^2)^2} = 0 \tag{E.2}$$

Thus, the value of ρ that maximizes the loss satisfies:

$$-2\delta(1-\rho^2) + 2\rho(\delta^2 - 2\rho\delta + 1) = 0 \tag{E.3}$$

The above equation has two solutions, $\rho = 1/\delta$ and $\rho = \delta$, and it can be verified that the first corresponds to a maximum of the loss function $L(h^{(B)})$ while the second corresponds to a local minimum. Since $\rho \in [-1, 1]$ and $\delta \geq 1$, we focus on the upper limit of the error, which occurs when $\rho = 1/\delta$. Thus,

$$0 \leq L(h^{(B)}) \leq 1 - \Phi(1). \tag{E.4}$$

The function Φ approaches 1 when its argument tends to infinity. The lower limit of the error is reached for $\rho \approx -1$, corresponding to scenarios where X_2 is a deterministic function of X_1.

As ρ increases from -1 to $1/\delta$, the error increases; as ρ increases further, the error decreases to zero.

F FOUR SCENARIOS OF INTEREST IN THE 3D CASE

This appendix is a continuation of Appendix D.3.

F.1 INTRODUCTION

In this appendix, we consider the classification problem in the three-dimensional case.

- The conditional distributions of X, given the label Y, are both trivariate normal distributions.
- The two centroids are fixed at $(1, 1, 1)$ and $(-1, -1, -1)$.
- The covariance matrix Σ, which depends on 6 parameters, is the same for the two Gaussians considered, one for each class.
- For a given problem, defined by the matrix Σ, once the sample size n is fixed, there is no theoretical difficulty in programming the simulator to calculate the expected error rate.

Why were the 4 scenarios created?

- The main theoretical difficulty to be resolved is determining the optimal separating plane, whose equation we represent by: $x_3 = d^* + e^* x_1 + f^* x_2$. Once this equation is obtained, the corresponding Bayes Risk can be automatically calculated.
- Given the symmetry of the problem with respect to the origin $(0, 0, 0)$, the optimal separating plane must necessarily pass through the origin. In other words, we know that $d^* = 0$.
- Thus, we are left with a minimization problem with two unknowns: e^* and f^*.
- We have attempted to obtain an analytical solution for this problem, i.e., to express e^* and f^* mathematically as functions of the 6 parameters that define the matrix Σ. So far, we have not been successful.
- This is where the 4 scenarios come in. In each of them, simplifying assumptions about the matrix Σ are introduced, so that it depends on only 3 parameters. Under these conditions, we can obtain analytical solutions to the problem.
- Another possible alternative, which would eliminate the need for the 4 scenarios, would be to attempt to solve this minimization problem using numerical analysis. We leave this approach as a topic for future work.

F.2 FOUR SCENARIOS

To simplify the mathematics, we will consider four specific scenarios. In each scenario, the idea is to reduce the number of free parameters, allowing analytical expressions to be obtained for the optimal coefficients of the plane, e^* and f^*, and the Bayes risk.

In what follows, when we say that two attributes are independent, we are referring to conditional independence, given that the class of the observation is known. Similarly, when we say that two attributes are correlated, we are referring to conditional correlation, given that the class of the observation is known.

1. **Scenario 1:** The 3 attributes are pairwise independent, i.e., $\rho_{12} = \rho_{13} = \rho_{23} = 0$. In this case, it can be shown that:

   a) $e^* = -\frac{\sigma_3^2}{2\sigma_1^2}$

   b) $f^* = -\frac{\sigma_3^2}{2\sigma_2^2}$

   c) $P(\text{Error}) = 1 - \phi\left(\frac{\sqrt{1-e^*-f^*}}{\sigma_3}\right)$

2. **Scenario 2:** X_1 and X_2 are correlated, $(X_1,\, X_2)$ and X_3 are independent

   - Attributes X_1 and X_2 are correlated, i.e., $\rho_{12} = \rho$ can be $\neq 0$
   - Attributes X_1 and X_3 are independent, i.e., $\rho_{13} = 0$
   - Attributes X_2 and X_3 are independent, i.e., $\rho_{23} = 0$
   - $\sigma_1 = \sigma_2 = \sigma$

   In this case, it can be shown that:

   a) $e^* = f^* = -\frac{\sigma_3^2}{2\sigma^2(1+\rho)}$

   b) $P(\text{Error}) = 1 - \phi\left(\sqrt{1 - 2e^*\frac{1}{\sigma_3}}\right)$

3. **Scenario 3:** X_1 and X_2 are independent, (X_1, X_2) and X_3 are correlated

   - Attributes X_1 and X_2 are independent, i.e., $\rho_{12} = 0$
   - Attributes X_1 and X_3 are correlated, i.e., ρ_{13} can be different from 0
   - Attributes X_2 and X_3 are correlated, i.e., ρ_{23} can be different from 0
   - $\rho_{13} = \rho_{23} = r$ (Restriction: $-\frac{\sqrt{2}}{2} < r < \frac{\sqrt{2}}{2}$)
   - $\sigma_1 = \sigma_2 = \sigma$

   In this scenario, it can be shown that:

   a) $e^* = f^* = \frac{\sigma_3}{(\sigma_3 - r\sigma)} \cdot \frac{\sigma}{(2r\sigma_3 - \sigma)}$

   b) $P(\text{Error}) = 1 - \phi\left(\frac{1-2e^*}{\sqrt{\Delta}}\right)$, where $\Delta = 2(e\sigma - r\sigma_3)^2 + \sigma_3^2(1 - 2r^2)$

4. **Scenario 4:** X_1 and X_2 are correlated, (X_1, X_2) and X_3 are correlated

   - All attributes have equal dispersion $(\sigma_1 = \sigma_2 = \sigma_3 = \sigma)$
   - Attributes X_1 and X_2 are correlated, i.e., $\rho_{12} = \rho$ can be different from 0
   - Attributes X_1 and X_3 are correlated, i.e., ρ_{13} can be different from 0
   - Attributes X_2 and X_3 are correlated, i.e., ρ_{23} can be different from 0
   - $\rho_{13} = \rho_{23} = r$
   - Restriction: $-\sqrt{\frac{1+\rho}{2}} \leq r \leq \sqrt{\frac{1+\rho}{2}}$

   In this scenario, it can be shown that:

   a) $e^* = f^* = \frac{1-r}{2r-(1+\rho)}$

   b) $P(\text{Error}) = 1 - \phi\left(\frac{1-2e^*}{\sqrt{\Delta}}\right), \quad \Delta = A^2 + B^2 + \lambda_{33}^2$

   where:

   - $A = \sigma[e^*(1+\rho) - r]$;
   - $B = \sigma\left[\sqrt{1-\rho^2}e^* - \frac{r(1-\rho)}{\sqrt{1-\rho^2}}\right]$;
   - $\lambda_{33} = \sigma\sqrt{1 - 2r^2/(1+\rho)}$

G A SIMPLE BISECTOR FOR TWO OBSERVATIONS ($n = 2$) WITH ONE OR TWO ATTRIBUTES

The results from the previous section are asymptotic results for an infinite sample size. Now, we consider the extreme opposite: a dataset composed of two samples, one from each class. We denote by $h_1^{(SVM)}$ and $h_2^{(SVM)}$ the classifier obtained using **SVM**, with one attribute (X_1) and two attributes, respectively. Thus, the **SVM** solutions correspond to simple bisectors as separation boundaries. The expected errors are given by

$$E\left(L(h_1^{(SVM)})\right) = \int_{c=-\infty}^{+\infty} L_1(c)\frac{1}{\sqrt{\pi}}e^{-c^2}dc \approx 0.2070336 \tag{G.1}$$

and

$$E\left(L(h_2^{(SVM)})\right) = \int_{\boldsymbol{w}\in\mathbb{R}^2}\int_{\boldsymbol{v}\in\mathbb{R}^2} L_2(g(\boldsymbol{v},\boldsymbol{w}))\phi_{+1}(\boldsymbol{v})\phi_{-1}(\boldsymbol{w})d\boldsymbol{v}d\boldsymbol{w} \tag{G.2}$$

where the expectations are taken over random training sets with two observations each, drawn from bivariate Gaussians, $\boldsymbol{v} = (v_1, v_2)$, $\boldsymbol{w} = (w_1, w_2)$, $\phi_t(\cdot)$ is the probability density function of the bivariate Gaussian corresponding to class t, and $g(\boldsymbol{v},\boldsymbol{w})$ is a function that maps a pair of points to their intersection and the slope of the perpendicular bisector.

$$g(\boldsymbol{v},\boldsymbol{w}) = \left(\frac{w_1 - v_1}{w_2 - v_2}\cdot\frac{v_1 + w_1}{2} + \frac{v_2 + w_2}{2}, -\frac{w_1 - v_1}{w_2 - v_2}\right). \tag{G.3}$$

Comparing $L_1(0)$ with $L(h_1^{(SVM)})$, (D.2) with (G.1), we evaluate the impact of finite sample size when only one attribute is available and compare the performance of the loss function $L(h^{(B)})$ with that of the loss function $L(h_2^{(SVM)})$, (D.23) with (G.2), when two attributes are available.

For $\rho = 0$, the corresponding loss values are presented in the rows $n = 1,024$ and $n = 2$ of Table 1, whose values agree up to three decimal places with the expressions (G.1)-(G.2) and (D.2)-(D.23), respectively. Comparing (G.1) with (G.2), which correspond to the elements in the first row of Table 1, it is noted that when the sample size is small and $\delta \geq 2$, it is beneficial to use fewer attributes. On the other hand, comparing (D.2) with (D.23), which correspond to the elements in the last row of Table 1, it is noted that when the sample size is small and $1 \leq \delta \leq 7$, it is beneficial to use more attributes.

H A MODEL TO UNDERSTAND THE BEHAVIOR OF BAYES RISK

H.1 ONE AND TWO DIMENSIONS

Suppose that only X_1 and X_2 are present. Consider an ellipse in $\mathbb{R}^2$ (say, centered at the origin $(0,0)$) whose equation is

$$x^T\Sigma^{-1}x = 1.$$

For each class, we have an ellipse, and for each ellipse, we have a centroid. The line that connects the two centroids $(1,1)$ and $(-1,-1)$ is the set of all vectors in $\mathbb{R}^2$ of the type

$$(c,c),$$

whose two coordinates are equal. This line cuts through the ellipse $x^T\Sigma^{-1}x = 1$ at some point with equal and positive coordinates,

$$(c_2,c_2)\cdot\Sigma^{-1}\begin{pmatrix} c_2 \\ c_2 \end{pmatrix} = 1.$$

Let d_2 be the distance of this point from the origin,

$$d_2 = \sqrt{2}c_2.$$

Intuitively, we notice that when this distance is large (Figure 29(a)), the ellipse aligns with the line that connects the centroids, making classification more difficult compared to scenarios where the distance is smaller (Figure 29(b)).

We do not have a formal argument to identify the conditions under which an increase in distance d_2 implies a decrease in classification power, but we have verified numerically that in several scenarios, the larger this distance, the lower the discrimination power of the classifier when both X_1 and X_2 are present. In particular, comparing Figure 29(a) with Figure 29(b), we find that classifying points in Figure 29(b) is easier than classifying points in Figure 29(a), bearing in mind that both clouds of points have centroids $(+1,+1)$ and $(-1,-1)$, and both clouds obey the same covariance matrix. In other words, both clouds will resemble the pattern in Figure 29(a), centered at $(+1,+1)$ and $(-1,-1)$, or both clouds will resemble the pattern in Figure 29(b), centered at $(+1,+1)$ and $(-1,-1)$. Classifying in the first case is more difficult than in the second, as illustrated in Figure 30.

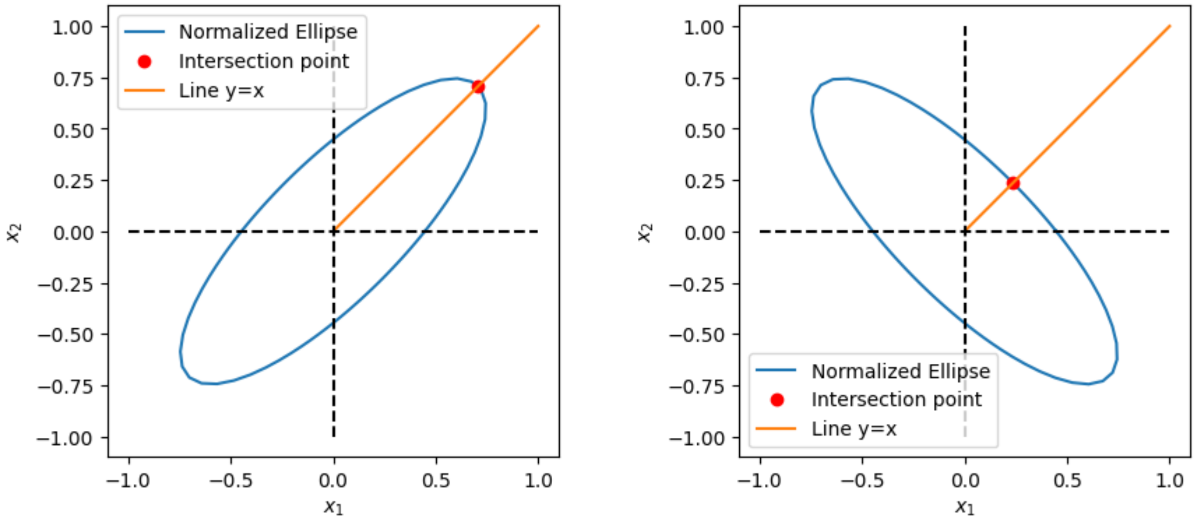
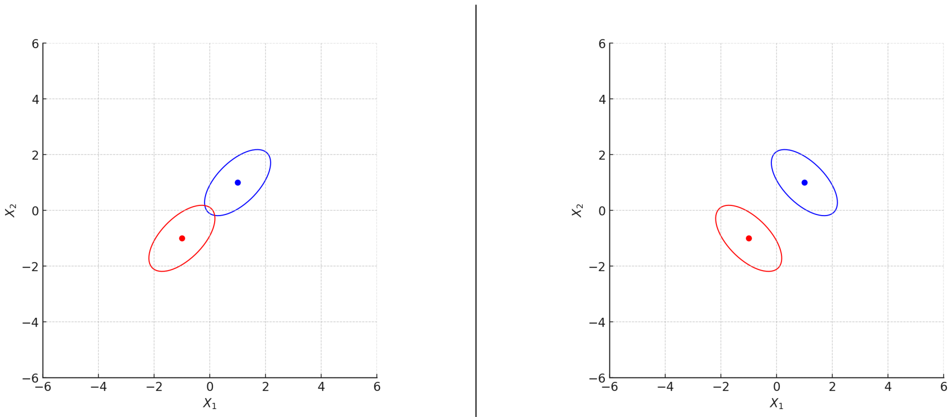


Figure 29 – d_2 for $\sigma_1 = 1, \sigma_2 = 1, \rho = 0.8$ and d_2 for $\sigma_1 = 1, \sigma_2 = 1, \rho = -0.8$



(a)

(b)

Figure 30 – Classifier with more (a) difficult ($\rho = 0.8$) and (b) easy ($\rho = -0.8$) tasks.

H.2 EXTENDING TO THE THREE-DIMENSIONAL CASE

Extending the discussion, consider an ellipsoid in $\mathbb{R}^3$ (say, centered at the origin $(0, 0, 0)$) whose equation is

$$x^T \Sigma^{-1} x = 1,$$

where now $x \in \mathbb{R}^3$.

Recall that each class of points (e.g., orange and blue) is associated with an ellipsoid, which contains a centroid. The line connecting the two centroids is the set of all vectors in $\mathbb{R}^3$ of the type (c, c, c), with all three coordinates equal. This line intersects the ellipsoid at a point with all three coordinates equal and positive.

$$(c_3, c_3, c_3) \cdot \Sigma^{-1} \begin{pmatrix} c_3 \\ c_3 \\ c_3 \end{pmatrix} = 1.$$

Let d_3 be the distance from this point to the origin,

$$d_3 = \sqrt{3} c_3.$$

Intuitively and numerically, we have verified in numerous scenarios that the larger this distance, the lower the discrimination power of the classifier when X_1, X_2, and X_3 are present. We leave a formal verification of the conditions under which this relationship applies for future work.

Below, we indicate how d_2 and d_3, which can be easily calculated using scientific computing software, help determine the utility of the attribute X_3.

H.3 COMPARING THE TWO AND THREE ATTRIBUTE SCENARIOS

How can we use d_3 and d_2 to create, from the matrix Σ, an indicator of the additional discrimination potential of X_3, given that X_1 and X_2 are already present? The indicator we propose is d_2/d_3,

$$U_3(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \frac{d_2}{d_3} = \sqrt{\frac{2}{3}} \frac{c_2}{c_3}, \tag{H.1}$$

where U_3 is a utility metric for the third attribute.

Let R_2 be the Bayes risk with two attributes, and R_3 the corresponding risk with three attributes. Empirically, we verify that the ratio between the risks is a smooth and increasing function of the above utility function, i.e.,

$$\frac{R_2}{R_3} = \varphi(U_3), \tag{H.2}$$

where the function $\varphi(\cdot)$ is a smooth, increasing function, to be experimentally determined.

H.4 EVALUATING THE UTILITY OF THE THIRD ATTRIBUTE

To evaluate the utility of the third attribute, X_3, given the attributes X_1 and X_2, we consider the following plots as outputs from our simulator for a given scenario:

- $\log_2 n^*$ as a function of the values of the considered parameter α: recall that the threshold $n^\star$ corresponds to the number of observations beyond which it becomes advantageous to include attribute X_3, given that the other two attributes X_1 and X_2 are already present. The smaller the value of $n^\star$, the greater the range justifying the inclusion of X_3. For more details, see Section 3.1;

- $\log_2 n^*$ **vs** R_2/R_3: while the above-described plot has the parameter α values on the x-axis, in this and the next plot, we consider alternative metrics that somehow capture the utility of attribute X_3 for each value of the considered parameter α. Firstly, we consider the ratio between the Bayes risk in the case of having two attributes (X_1, X_2) and in the case of having three attributes (X_1, X_2, X_3). Our goal here is to explain what affects the value of $n^\star$. In particular, we aim to understand to what degree the Bayes risk can be used to justify the behavior of $n^\star$. For more details, see Section 3.1 and Appendix D;

- $\log_2 n^*$ **vs** U_3: once again, our goal here is to explain what affects the value of $n^\star$. To this end, we propose a new metric, U_3, and present plots of $\log_2 n^\star$ as a function of U_3. For more details, see Appendix H.3;

- U_3 **vs** R_2/R_3: as indicated above, we have two attempts to explain the behavior of $n^\star$, the first based on Bayes risk and the second based on a new proposed metric, U_3. In this final plot, we relate these two attempts to explain the behavior of $n^\star$ by presenting one against the other. For more details, see Appendix D and Appendix H.3.