Minimally Informed Linear Discriminant Analysis: training an LDA model with unlabelled data*

Nicolas Heintz[a,b], Tom Francart[b], Alexander Bertrand[a]

[a] *KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Leuven, Belgium*
[b] *KU Leuven, Department of Neurosciences, ExpORL, Leuven, Belgium*

Abstract

Linear Discriminant Analysis (LDA) is one of the oldest and most popular linear methods for supervised classification problems. Computing the optimal LDA projection vector requires calculating the average and covariance of the feature vectors of each class individually, which necessitates class labels to estimate these statistics from the data. In this paper we demonstrate that, if some minor prior information is available, it is possible to compute the exact projection vector from LDA models based on unlabelled data. More precisely, we show that either one of the following three pieces of information is sufficient to compute the LDA projection vector if only unlabelled data are available: (1) the class average of one of the two classes, (2) the difference between both class averages (up to a scaling), or (3) the class covariance matrices (up to a scaling). These theoretical results are validated in numerical experiments, demonstrating that this minimally informed Linear Discriminant Analysis (MILDA) model closely approximates the solution of a supervised LDA model, even on high-dimensional, poorly separated or extremely imbalanced data. Furthermore, we show that the MILDA projection vector can be computed in a closed form with a computational cost comparable to LDA and is able to quickly adapt to non-stationary data, making it well-suited to use as an adaptive classifier that is continuously retrained on (unlabelled) streaming data.

Keywords: unsupervised learning, semi-supervised learning, classification, linear discriminant analysis, adaptive learning

1. Introduction

Fisher's Linear Discriminant Analysis (LDA) model is a standard algorithm in data science for linear classification [1]. Its goal is to find the optimal linear projection of the data such that two classes are maximally separated. Despite the emergence of more advanced techniques, LDA remains a very popular data analysis tool across scientific fields [2, 3, 4, 5, 6, 7, 8] because it is easy to understand, it can be trained on very few data points, it is cheap to compute, and it has an elegant and interpretable formulation rooted in linear algebra. Its low computational complexity also makes it a popular tool for signal processing, in particular in adaptive settings and in real-time or low-resource environments [3, 4].

In his original LDA paper, Fisher proposed a transformation that maximises the distance between class averages and minimises the variance within each class [9]. Computing Fisher's LDA model requires four (or two) parameters: the two class averages (or their difference) and the class covariance matrices of both classes

*This research is funded by Aspirant Grant 1S31522N (for N. Heintz) from the Research Foundation - Flanders (FWO), the Research Foundation - Flanders (FWO) project No G081722N, Internal Funds KU Leuven (project/IDN/23/006), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 802895 and No 101138304), and the Flemish Government (AI Research Program). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the granting authorities. Neither the European Union nor the granting authorities can be held responsible for them. N. Heintz, T. Francart and A. Bertrand are also affiliated with Leuven.AI - KU Leuven institute for AI, Belgium. N. Heintz and A. Bertrand are EURASIP members.

Published in Signal Processing, Volume 239, 2026, 110226, ISSN 0165-1684, https://doi.org/10.1016/j.sigpro.2025.110226

(or their sum). These parameters can either be known via prior knowledge or estimated using labelled data. Herein lies one of the main drawbacks of LDA (and any other supervised model): as soon as one of these parameters is unknown, a labelled dataset is required to train the LDA model. For many applications, this requires manual labelling by experts, which is time-consuming and costly. Furthermore, in a non-stationary setting, i.e. when the statistics of the data change over time, it is often important to regularly re-train the model. This only exacerbates the aforementioned issues, as labels are typically unavailable during real-time operation of the algorithm.

These drawbacks have motivated researchers to develop unsupervised methods that can approximate the LDA projection vector. It was shown in [10] that the vector maximising the absolute value of the kurtosis coefficient of the projections is equivalent to the LDA projection vector if the distributions are symmetric, have proportional covariance matrices and are elliptical. Similarly, the projection vector maximising skewness is proportional to the LDA projection vector if the class distributions are symmetrical, and if there are significantly more samples of one class than another class [11]. However, these projections are relatively expensive to compute [11], fail at specific class imbalance ratios and are sensitive to outliers.

Alternatively, one can resort to clustering algorithms such as, e.g., Gaussian Mixture Models (GMMs) [12], K-means [13], and its extension K-means++ [14]. In some cases, the simplicity, modelling power and robustness of LDA have motivated the incorporation of LDA in such unsupervised frameworks. A popular approach is to iteratively estimate a set of pseudo-labels using some unsupervised clustering method and then train the LDA model with these labels [15, 16, 17].

The aforementioned methods allow computing a proxy of the LDA solution in cases where the class averages and covariances are unknown and cannot be estimated from the data due to the absence of labels in the training data. However, it is possible that some -but not all- of the relevant ground-truth statistics of the data are known, for example:

- Only the **class average** of a **single class** is known: This could occur, e.g., when there is a (partial) model for one of the classes, but not for the other. One example could be a detection problem where the noise is known to be zero-mean, but where its covariance structure is unknown. In the absence of a detection target (class 0), the class average is then known to be 0. Another example is anomaly detection, where the mean of the 'normal' class is generally known or can be estimated from available data. Finally, class averages can typically be estimated with much less data than covariances matrices. This is particularly relevant for semi-supervised learning, where only a small partition of the dataset is labelled.

- Only the **relative difference** between the **class averages** is known (up to a scaling): This is relevant when it is known in what direction the features are expected to move when shifting from one class to the other, without knowing how large the change is expected to be, what the exact class averages are or what variance is expected. This is the case, for example, when the amplitude changes in an amplitude-modulated signal, which must be detected with a sensor array. The steering vector for the target source is often known up to an unknown scalar, while there can simultaneously be ambiguity in the amplitude of the source signal.

- Only the **covariances** for each of the **two classes** are known (up to a scaling): Such a scenario is common when, e.g., the noise covariance statistics are known (possibly up to an unknown calibration factor). This also includes the common case where the noise is independent and identically distributed across the feature space.

To the best of the authors' knowledge, there exists no method to compute the LDA solution where only one of the three aforementioned statistics are known (remember that LDA requires the per-class average and covariance matrices). Only in the last setting was it shown in [18] that the LDA solution can be approximated for the case where the two classes are sufficiently separated and have a Gaussian distribution.

In this paper, we present a mathematical formulation that makes it possible to compute the LDA projection with only *one* of the aforementioned ground-truth statistics, which we refer to as 'minimally informed' LDA (MILDA). In contrast to [18], MILDA is applicable if any of the aforementioned ground-truth statistics are known ([18] only covers the case with known covariances), while we do not require

Gaussian class distributions nor a good class separability. In fact, we will show that MILDA remains stable even in problems with poor class separability, whereas traditional unsupervised clustering methods generally fail in such scenarios.

The proposed MILDA framework still results in closed-form expressions, which are elegant and cheap to compute. This is in line with the main advantages of the standard LDA. MILDA optimises a slightly more general objective function than LDA, and has the original LDA objective as a special case when either the classes are balanced or if the homoscedasticity assumption is satisfied. However, even if neither of these assumptions hold, MILDA empirically still achieves similar accuracies as LDA. Furthermore, MILDA becomes more robust to estimation errors in the ground-truth statistic when the classes are less separated or imbalanced. This is in stark contrast with most unsupervised clustering methods.

The aforementioned properties make MILDA especially suited for use in adaptive modelling. Typical examples are, e.g., classifying a non-stationary signal with stationary noise or vice versa. The time-independent statistics can then be estimated and used as ground-truth statistics, while MILDA automatically adapts to the time-varying statistics. Finally, MILDA is ideally suited to classify heavily imbalanced datasets. We will show that just a few labelled samples of the minority class suffice to train a MILDA model that closely approximates the performance of a linear model trained on the entire dataset.

The outline of the paper is as follows. In Section 2, we describe the main mathematical framework behind the MILDA model. In Section 3 we provide a theoretical analysis of the sensitivity of MILDA to errors in the used ground-truth statistic. In Section 4, we benchmark MILDA on simulated data in an ablation study. Section 5 focuses on a few use cases for MILDA, such as target signal detection, adaptive MILDA for non-stationary data, evoked potential detection for brain-computer interfaces and credit card fraud detection. Finally, in Section 6, we summarise the main conclusions from this paper and discuss possibilities for future work.

As a notational convention, we use regular letters (a) to denote scalars, bold letters ($\mathbf{a}$) for vectors, capital letters (A) for constants and matrices (depending on the context) and Greek letters (α) for parameters and latent scalars to shorten equations.

2. Minimally informed LDA (MILDA)

In this section, we will first briefly review the main mathematical framework of LDA as proposed by Fisher [9]. In Section 2.2, we show how the same projection can be found with the temporary (and overly strict) assumption that the two class averages are proportionate to each other. This generates an intermediate result that will be needed in the derivation of MILDA. Since this assumption of proportionality is rarely valid in real-life applications, we show in Section 2.3 how this assumption can be dropped when only a single ground-truth statistic is known. No other assumptions on the data are necessary to prove that the MILDA and LDA projections are equivalent.

2.1. Linear Discriminant Analysis

Let $\{\mathbf{x}_i\}_i$ with $\mathbf{x}_i \in \mathbb{R}^D$ be a set of D-variate samples, where each sample belongs to one of two classes. The goal of Linear Discriminant Analysis (LDA) is to find a projection vector $\mathbf{w} \in \mathbb{R}^D$ that maximally separates two classes after projecting each data point onto $\mathbf{w}$. An intuitive way to do this is by searching for the projection that maximises the distance between the class averages and minimises the variance within each class. By writing these two joint objectives as a ratio, Fisher proposed the following objective:

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top (\boldsymbol{\mu}_+ \text{-} \boldsymbol{\mu}_-)}{\mathbf{w}^\top (\Sigma_+ + \Sigma_-)\, \mathbf{w}}, \tag{1}$$

where $\boldsymbol{\mu}_+$, $\boldsymbol{\mu}_-$ are the class averages and Σ_+, Σ_- are the covariance matrices for each class [9]. If a labelled set of samples $\mathbf{x}_i$ is available, then the class averages and class covariances can be estimated as[1]:

$$\boldsymbol{\mu}_\pm \triangleq \frac{1}{N_\pm} \sum_{i \in \mathcal{C}_\pm} \mathbf{x}_i \tag{2}$$

$$\Sigma_\pm \triangleq \frac{1}{N_\pm} \sum_{i \in \mathcal{C}_\pm} (\mathbf{x}_i - \boldsymbol{\mu}_\pm)(\mathbf{x}_i - \boldsymbol{\mu}_\pm)^\top, \tag{3}$$

where $\mathcal{C}_+$ and $\mathcal{C}_-$ contain the sample indices of the positive and negative class, respectively, with cardinalities $N_+ = |\,\mathcal{C}_+\,|$ and $N_- = |\,\mathcal{C}_-\,|$. The solution of (1) is then given by:

$$\mathbf{w}^* \propto (\Sigma_+ + \Sigma_-)^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-), \tag{4}$$

assuming $\Sigma_+ + \Sigma_-$ is full rank[2] [9]. Here we use the notation $\propto$ to denote that the left- and right-hand sides are equal up to an arbitrary non-zero scaling.

It was later shown that this solution minimises the classification error if both classes have a Gaussian distribution and if they are homoscedastic ($\Sigma_+ = \Sigma_-$), i.e., no other classifier can result in better classification performance [1]. In this case, $\mathbf{w} \propto \Sigma^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$, with $\Sigma = \Sigma_+ = \Sigma_-$. Confusingly, this solution is often also referred to as *Linear Discriminant Analysis*, but it is an edge case of the more general LDA formulation provided by Fisher. We note that Gaussianity and homoscedasticity should not be viewed as limiting assumptions in order to use LDA, i.e., without these assumptions, the LDA objective (1) remains relevant and often results in a good classifier [2, 1].

In this paper, we will show that MILDA provides a projection vector that optimises a slightly more general objective (compared to (1)):

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)}{\mathbf{w}^\top(q\Sigma_+ + (1-q)\Sigma_-)\,\mathbf{w}}, \tag{5}$$

with $q = N_+/N$ the fraction of samples in a dataset that belong to the positive class and $N = N_+ + N_-$ the total number of samples in a class. The solution of (5) is given by:

$$\mathbf{w}^* \propto \hat{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-), \tag{6}$$

$$\hat{\Sigma} \triangleq q\Sigma_+ + (1-q)\Sigma_-. \tag{7}$$

Note that (6) is identical to the original LDA solution (4) if either $\Sigma_+ \propto \Sigma_-$ (e.g. when the classes are homoscedastic) or if $q = 0.5$, meaning that the classes are balanced. When this is not the case, MILDA will put more weight on minimising the class variance of the class containing more samples compared to the class containing fewer samples than LDA. This could theoretically be beneficial when little data are available, as in that case, the covariance matrices are sensitive to estimation errors. However, in practice we found no noteworthy difference in performance between (6) and (4), as demonstrated in Section 4.

2.2. Preliminaries to MILDA

The projection from (6) requires knowledge of Σ_+, Σ_-, the class difference $\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$, and q. In practice, these parameters can usually only be estimated on labelled data in a supervised setting. However, in this section we will show that the solution (6) can be computed without labels if the two class averages are proportional[3], i.e., $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$. While the latter assumption may seem artificial and overly strict at first sight, we will show in Section 2.3 that it can be relaxed to more realistic conditions based on knowledge of some ground-truth statistics.

[1] We use the biased covariance estimator throughout this paper for notational convenience.

[2] If the summed covariance matrix is rank-deficient, a dimensionality reduction should be performed first, e.g., via principal component analysis.

[3] We use $\mathbf{x} \sim \mathbf{y}$ to denote that either $\mathbf{x} = \alpha\mathbf{y}$ or $\mathbf{y} = \beta\mathbf{x}$ where $\alpha, \beta \in \mathbb{R}$, whereas we use $\mathbf{x} \propto \mathbf{y}$ to denote that $\mathbf{x} = \alpha\mathbf{y}$ with $\alpha \in \mathbb{R}\setminus\{0\}$. In the former case, it is allowed that either $\mathbf{x}$ or $\mathbf{y}$ is the zero vector, whereas in the latter both vectors should be non-zero.

Lemma 2.1. *If $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$, then the solution of (5) can be computed as*

$$\mathbf{w}^* \propto \bar{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}} \tag{8}$$

$$\textit{with}\ \bar{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{9}$$

$$\bar{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\boldsymbol{\mu}})(\mathbf{x}_i - \bar{\boldsymbol{\mu}})^\top \tag{10}$$

Proof. With some algebraic manipulations, we can rewrite the covariance matrix $\bar{\Sigma}$ in (10) and class covariance matrices Σ_+ and Σ_- in (3) as:

$$\bar{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^\top - \bar{\boldsymbol{\mu}}\,\bar{\boldsymbol{\mu}}^\top \tag{11}$$

$$\Sigma_+ = \frac{1}{qN}\sum_{i\in\mathcal{C}_+}\mathbf{x}_i\mathbf{x}_i^\top - \boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^\top \tag{12}$$

$$\Sigma_- = \frac{1}{(1-q)N}\sum_{i\in\mathcal{C}_-}\mathbf{x}_i\mathbf{x}_i^\top - \boldsymbol{\mu}_-\,\boldsymbol{\mu}_-{}^\top\,. \tag{13}$$

Based on (7), (11)-(13), it can be verified that:

$$\begin{aligned}\bar{\Sigma} &= q(\Sigma_+ + \boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^\top) + (1-q)(\Sigma_- + \boldsymbol{\mu}_-\,\boldsymbol{\mu}_-{}^\top) - \bar{\boldsymbol{\mu}}\,\bar{\boldsymbol{\mu}}^\top \\ &= \hat{\Sigma} + q\,\boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^\top + (1-q)\,\boldsymbol{\mu}_-\,\boldsymbol{\mu}_-{}^\top - \bar{\boldsymbol{\mu}}\,\bar{\boldsymbol{\mu}}^\top\,.\end{aligned} \tag{14}$$

Furthermore, from (2) and (9), we can also write

$$\bar{\boldsymbol{\mu}} = q\,\boldsymbol{\mu}_+ + (1-q)\,\boldsymbol{\mu}_-\,. \tag{15}$$

Since $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$, we define[4] $\boldsymbol{\mu}_- = \alpha\,\boldsymbol{\mu}_+$ with $\alpha \in \mathbb{R}\setminus\{1\}$. Note that $\alpha = 1$ is excluded, since this would imply that the mean of both classes coincides, in which case (5) is ill-defined. By substituting $\boldsymbol{\mu}_- = \alpha\,\boldsymbol{\mu}_+$ in (14) and (15), we can simplify $\bar{\boldsymbol{\mu}}$ and $\bar{\Sigma}$ as:

$$\begin{aligned}\bar{\boldsymbol{\mu}} &= (q + (1-q)\alpha)\,\boldsymbol{\mu}_+ && \triangleq \beta\,\boldsymbol{\mu}_+ \\ \bar{\Sigma} &= \hat{\Sigma} + (q + (1-q)\alpha^2 - \beta^2)\,\boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^\top && \triangleq \hat{\Sigma} + \gamma\,\boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^\top,\end{aligned}$$

with $\gamma = q(1-q)(\alpha-1)^2 > 0$.

When plugging this in the right-hand side of (8), we get.

$$\begin{aligned}\mathbf{w}^* &\propto \bar{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}} \\ &\propto (\hat{\Sigma} + \gamma\,\boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^\top)^{-1}\,\boldsymbol{\mu}_+,\end{aligned} \tag{16}$$

where β is absorbed in the $\propto$ sign.

We will now show that (16) is indeed a solution of (5). First, the matrix inverse in this expression can be rewritten using the Sherman-Morrison identity, which is a special case of the Woodbury identity for rank-1

[4] If $\boldsymbol{\mu}_+ = \mathbf{0}$, the roles of $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$ should be switched in the proof.

updates [19]. This allows us to write (16) as:

$$\begin{aligned}\mathbf{w}^* &\propto \left(\hat{\Sigma}^{-1} - \frac{\gamma\,\hat{\Sigma}^{-1}\boldsymbol{\mu}_+\boldsymbol{\mu}_+{}^\top\hat{\Sigma}^{-1}}{1+\gamma\,\boldsymbol{\mu}_+{}^\top\hat{\Sigma}^{-1}\boldsymbol{\mu}_+}\right)\boldsymbol{\mu}_+ \\ &\propto \hat{\Sigma}^{-1}\boldsymbol{\mu}_+ - \frac{\gamma\,\boldsymbol{\mu}_+{}^\top\hat{\Sigma}^{-1}\boldsymbol{\mu}_+}{1+\gamma\,\boldsymbol{\mu}_+{}^\top\hat{\Sigma}^{-1}\boldsymbol{\mu}_+}\hat{\Sigma}^{-1}\boldsymbol{\mu}_+ \\ &\propto \hat{\Sigma}^{-1}\boldsymbol{\mu}_+, \end{aligned} \tag{17}$$

where the irrelevant scaling factor between the left- and right-hand side were absorbed in the $\propto$ sign.

Finally, since $\boldsymbol{\mu}_+ - \boldsymbol{\mu}_- = (1-\alpha)\,\boldsymbol{\mu}_+ \propto \boldsymbol{\mu}_+$, we can rewrite (17) as:

$$\mathbf{w}^* \propto \hat{\Sigma}^{-1}(\boldsymbol{\mu}_+ \text{-} \boldsymbol{\mu}_-).$$

This results in (6), which is a solution of (5), proving the lemma. □

As stated before, the vector $\mathbf{w}^*$ in (6) or (8) is proportional to Fisher's LDA defined in (4) if the classes are balanced ($q = 0.5$) or if $\Sigma_+ \propto \Sigma_-$. Otherwise, this vector optimises the objective formulated in (5).

2.3. The MILDA framework

Section 2.2 showed that the LDA projection vector can be computed using the label-free expression in (8) if the two class averages are proportional to each other or if either class is zero-mean, i.e. if $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$. However, in practice this rather artificial assumption is rarely satisfied. Fortunately, it is possible to satisfy this assumption by transforming the data using one out of the three possible ground-truth statistics: the class average of a single class, the relative difference between the class averages (up to a scaling) or the covariances for each of the two classes (up to a scaling). The latter case also assumes either balanced classes (or knowledge of the class imbalance), or a relaxed homoscedasticity condition (i.e. $\Sigma_+ \propto \Sigma_-$). We will go over these cases one by one and derive the required transformations. We will denote the transformed data and its corresponding statistics using the notation $(.)'$, i.e., the mean of the positive class after transformation is denoted as $\boldsymbol{\mu}'_+$. All transformations are summarised in Table 1 for convenience of the reader. The MILDA algorithm, which is summarised in Algorithm 1, then simply requires applying the corresponding transformation on the data and then computing (8).

2.3.1. The class average of a single class is known

Assume without loss of generality that the class average $\boldsymbol{\mu}_+$ is known. In this case, you can guarantee that $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$ after transforming each feature vector $\mathbf{x}_i$ to:

$$\mathbf{x}'_i \longleftarrow \mathbf{x}_i - \boldsymbol{\mu}_+\,. \tag{18}$$

Indeed, in this case $\boldsymbol{\mu}'_+ = \mathbf{0}$. Therefore $\boldsymbol{\mu}'_+ = 0\boldsymbol{\mu}'_-$, making $\boldsymbol{\mu}'_+ \sim \boldsymbol{\mu}'_-$ trivially true. The new global average $\bar{\boldsymbol{\mu}}' = \bar{\boldsymbol{\mu}} - \boldsymbol{\mu}_+ \neq \mathbf{0}$ as long as $\boldsymbol{\mu}_- \neq \boldsymbol{\mu}_+$, which ensures that the MILDA transformation from (8) has a solution. If $\boldsymbol{\mu}_- = \boldsymbol{\mu}_+$, the two classes are not separable with either LDA or MILDA and another classifier should be used.

2.3.2. The relative difference between the class averages is known

In this case, $\boldsymbol{\mu}_+ - \boldsymbol{\mu}_- = \delta\mathbf{d}$, with $\delta \in \mathbb{R}$ an unknown scalar and $\mathbf{d}$ a known vector. It is thus unknown by how much the classes are separated. We can guarantee that $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$ by transforming each data sample $\mathbf{x}_i$ to:

$$\mathbf{x}'_i \longleftarrow \mathbf{x}_i - (\bar{\boldsymbol{\mu}} - \alpha\mathbf{d}), \alpha \neq 0. \tag{19}$$

Though parameter $\alpha \neq 0$ can be freely chosen, it is best to choose α such that $\|\alpha\mathbf{d}\| \approx \|\mathbf{x}_i\|$ to avoid ill-conditioning.

After the transformation and using (15), we can write the new class averages as:

$$\begin{aligned}\boldsymbol{\mu}'_+ &= \boldsymbol{\mu}_+ - \bar{\boldsymbol{\mu}} + \alpha\mathbf{d} = (1-q)(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) + \alpha\mathbf{d} \\ &= ((1-q)\delta + \alpha)\mathbf{d} \\ \boldsymbol{\mu}'_- &= \boldsymbol{\mu}_- - \bar{\boldsymbol{\mu}} + \alpha\mathbf{d} = -q\,\boldsymbol{\mu}_+ + q\,\boldsymbol{\mu}_- + \alpha\mathbf{d} \\ &= (-q\delta + \alpha)\mathbf{d}.\end{aligned}$$

This shows indeed that $\boldsymbol{\mu}'_+ \sim \boldsymbol{\mu}'_-$. Furthermore, using (15) for the transformed data, we see that $\bar{\boldsymbol{\mu}}' \neq \mathbf{0}$ as long as $\alpha \neq 0$.

2.3.3. The covariances for each of the two classes are known

In this section, we assume that the covariances Σ_+ and Σ_- are known (up to a scaling), i.e., we assume the availability of the matrices $S_+ = \beta\Sigma_+$ and $S_- = \beta\Sigma_-$ with β an unknown scalar. In addition, we assume that the classes are balanced or that the class proportions (i.e. q) are known in case they are unbalanced. We will later see that this latter assumption can be removed if a relaxed homoscedasticity condition (i.e. $\Sigma_+ \propto \Sigma_-$) is satisfied.

We propose the following two-step transformation, which will later be shown (see Lemma 2.2) to result in $\boldsymbol{\mu}''_+ \sim \boldsymbol{\mu}''_-$:

$$\mathbf{x}'_i \longleftarrow \hat{S}^{-1/2}\mathbf{x}_i \tag{20}$$

$$\mathbf{x}''_i \longleftarrow \mathbf{x}'_i - (\bar{\boldsymbol{\mu}}' - \alpha\mathbf{d}), \alpha \neq 0, \tag{21}$$

$$\text{with} \quad \hat{S} = qS_+ + (1-q)S_- \propto \hat{\Sigma} \tag{22}$$

$$\bar{\boldsymbol{\mu}}' = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}'_i$$

$$\bar{\Sigma}' = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}'_i\mathbf{x}'^{\top}_i - \bar{\boldsymbol{\mu}}'\,\bar{\boldsymbol{\mu}}'^{\top}$$

$$\bar{\Sigma}'\,\mathbf{d} = \lambda_{max}\mathbf{d},$$

where λ_{max} is the largest eigenvalue of $\bar{\Sigma}'$. In other words, we first transform the data via (20). Then, we compute the eigenvector $\mathbf{d}$ that corresponds to the largest eigenvalue λ_{max} from the new covariance matrix $\bar{\Sigma}'$. In Lemma 2.2 below, we show that this vector $\mathbf{d}$ has the same direction as $\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-$. Therefore, once this direction is known, we can apply the same transformation as (19) to the data $\mathbf{x}_i$, resulting in (21).

Lemma 2.2. *Let $\bar{\Sigma}$ be defined as in* (11)*, and let $\hat{S}$ be defined as in* (22)*. If $\mathbf{d}$ is the eigenvector of $\bar{\Sigma}' = \hat{S}^{-1/2}\,\bar{\Sigma}\,\hat{S}^{-1/2}$ with the largest eigenvalue, then:*

$$\mathbf{d} \propto \hat{S}^{-1/2}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) = \boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-.$$

Proof. First, we will compute the effect of the first transformation on the covariance matrix. Substituting (15) in (14), we obtain

$$\begin{aligned}\bar{\Sigma} &= \hat{\Sigma} + q(1-q)\left(\boldsymbol{\mu}_+\,\boldsymbol{\mu}_+{}^{\top} + \boldsymbol{\mu}_-\,\boldsymbol{\mu}_-{}^{\top} - 2\,\boldsymbol{\mu}_+\,\boldsymbol{\mu}_-{}^{\top}\right) \\ &= \hat{\Sigma} + q(1-q)(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)^{\top}.\end{aligned} \tag{23}$$

After a transformation $\mathbf{x}'_i \longleftarrow A\mathbf{x}_i$, the new average and covariance are respectively $\bar{\boldsymbol{\mu}}' = A\,\bar{\boldsymbol{\mu}}$ and $\bar{\Sigma}' = A\,\bar{\Sigma}\,A^{\top}$. In this case, (23) becomes:

$$\begin{aligned}\bar{\Sigma}' &= A\,\hat{\Sigma}\,A^\top + q(1-q)(A\,\boldsymbol{\mu}_+ - A\,\boldsymbol{\mu}_-)(A\,\boldsymbol{\mu}_+ - A\,\boldsymbol{\mu}_-)^\top. \\ &\triangleq A\,\hat{\Sigma}\,A^\top + q(1-q)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)^\top. \end{aligned} \tag{24}$$

Since $\hat{S}$ is the sum of two symmetric positive definite matrices, it is also symmetric positive definite. Therefore, a symmetric matrix $\hat{S}^{-1/2}$ exists. From (22), we know that $\hat{S}^{-1/2} \propto \hat{\Sigma}^{-1/2}$. By defining $A = \hat{S}^{-1/2}$, (24) becomes:

$$\bar{\Sigma}' = \beta^{-1} I + q(1-q)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)^\top, \tag{25}$$

where β is an unknown scalar due to the proportionality $\hat{S} = \beta\,\hat{\Sigma}$. This shows that the transformation $\mathbf{x}'_i \longleftarrow \hat{S}^{-1/2}\mathbf{x}_i$ results in a covariance matrix that is equal to a rank-1 update on a scaled identity matrix. Note that if $\Sigma_+ \propto \Sigma_- \propto S$, the first transformation is proportional to $\mathbf{x}'_i \longleftarrow S^{-1/2}\mathbf{x}_i$, independent of q. Thus, knowledge of the class imbalance q is only required when $\Sigma_+ \not\propto \Sigma_-$.

Finally, we need to show that the vector $\mathbf{d} \propto \boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-$ is indeed the eigenvector of $\bar{\Sigma}'$ with the largest eigenvalue. By construction, the rank-1 matrix $B = q(1-q)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)^\top$ has only one eigenvector $\mathbf{d} \propto \boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-$ with a non-zero eigenvalue $\lambda_{B,1} = q(1-q)(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-)^\top(\boldsymbol{\mu}'_+ - \boldsymbol{\mu}'_-) > 0$. Note that $\bar{\Sigma}'$ is the matrix obtained by adding a scaled identity matrix $\beta^{-1}I$ to B. The impact of adding $\beta^{-1}I$ is that all eigenvalues are incremented with β^{-1}, while the eigenvectors remain unchanged. Therefore, the eigenvector $\mathbf{d}$ will still be the eigenvector of $\bar{\Sigma}'$ corresponding to the largest eigenvalue. □

Algorithm 1: A summary of the MILDA algorithm

Data: N samples $\mathbf{x}_i \in \mathbb{R}^{D\times 1}$
Result: Projection vector $\mathbf{w} \in \mathbb{R}^{D\times 1}$
$\mathbf{x}'_i \longleftarrow f(\mathbf{x}_i)$ as defined in Table 1
$\bar{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$,
$\bar{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^\top - \bar{\boldsymbol{\mu}}\,\bar{\boldsymbol{\mu}}^\top$
$\mathbf{w} \propto \bar{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}}$

Known	**Transformation** $f(\mathbf{x}_i)$
1) Class average $\boldsymbol{\mu}_+$	$\mathbf{x}'_i \longleftarrow \mathbf{x}_i - \boldsymbol{\mu}_+$
2) $\mathbf{d} \propto \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$	$\mathbf{x}'_i \longleftarrow \mathbf{x}_i - (\bar{\boldsymbol{\mu}} - \alpha\mathbf{d}), \alpha \neq 0$
3) $S_+ = \beta\Sigma_+,\ S_- = \beta\Sigma_-$ and q	$\mathbf{x}'_i \longleftarrow (qS_+ + (1-q)S_-)^{-1/2}\mathbf{x}_i$ $\mathbf{x}''_i \longleftarrow \mathbf{x}'_i - (\bar{\boldsymbol{\mu}}' - \alpha\mathbf{d}), \alpha \neq 0,$
4) $S \propto \Sigma_+ \propto \Sigma_-$	$\mathbf{x}'_i \longleftarrow S^{-1/2}\mathbf{x}_i$ $\mathbf{x}''_i \longleftarrow \mathbf{x}'_i - (\bar{\boldsymbol{\mu}}' - \alpha\mathbf{d}), \alpha \neq 0,$ with $\mathbf{d}$ defined in (21).

Table 1: A summary of the transformations required given the known piece of information.

2.4. Asymptotic computational complexity

Due to its close similarities to LDA, MILDA also has an asymptotic computational complexity of $\mathcal{O}(ND^2) + \mathcal{O}(D^3)$, with N the number of samples and D the data dimensionality. When MILDA is used in an adaptive context, its computational cost can be further reduced to $\mathcal{O}(D^2)$ per iteration by using similar techniques as used in, e.g., recursive least-squares (RLS) [1]. Since the asymptotic computational complexity is only relevant to measure how well the model scales, the average time required to classify N samples is also empirically measured in Section 4.8.

3. Sensitivity to errors on the known ground-truth statistic

Until now, we have always assumed that MILDA has access to the exact ground-truth statistic for each of the 4 cases in Table 1. In practice, often only an estimate of this statistic is available. In this section, we will therefore study how these approximation errors affect the projection vector from MILDA compared to the exact LDA projection vector (5). More precisely, we will study these effects both in a mathematical and experimental context in Sections 3.1 and 3.2 respectively. Amongst others, we will show that such approximation errors matter less when classes are more difficult to separate or more imbalanced, which are precisely the cases that are the hardest for unsupervised models to tackle.

3.1. Mathematical sensitivity analysis

Equation (8) states that MILDA computes the projection vector:

$$\mathbf{w}_{MILDA} \propto \bar{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}}, \tag{26}$$

with $\bar{\boldsymbol{\mu}}$, $\bar{\Sigma}$ defined as in (9),(10). Furthermore, from (23), we know that

$$\begin{aligned}\bar{\Sigma} &= \hat{\Sigma} + q(1-q)(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)^\top \\ &\triangleq \hat{\Sigma} + q(1-q)\,\boldsymbol{\mu}_\Delta\,\boldsymbol{\mu}_\Delta{}^\top,\end{aligned} \tag{27}$$

where we introduced a new notation $\boldsymbol{\mu}_\Delta \triangleq \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$ for notational convenience.

Using (27) in (26), applying the Sherman-Morrison identity [19] and rearranging the order of the terms gives:

$$\begin{aligned}\mathbf{w}_{MILDA} &\propto (\hat{\Sigma}^{-1} - \frac{q(1-q)\,\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta\,\boldsymbol{\mu}_\Delta{}^\top\,\hat{\Sigma}^{-1}}{1 + q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\,\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta})\,\bar{\boldsymbol{\mu}} \\ &= \hat{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}} - \frac{q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\,\hat{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}}}{1 + q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\,\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta}\,\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta \\ &= \hat{\Sigma}^{-1}(\bar{\boldsymbol{\mu}} + \phi\,\boldsymbol{\mu}_\Delta), \\ &\text{with } \phi \triangleq -\frac{q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\,\hat{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}}}{1 + q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\,\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta}.\end{aligned} \tag{28}$$

When comparing (28) with (6), we find that MILDA optimises (5) if and only if $\bar{\boldsymbol{\mu}}$ is proportional to $\boldsymbol{\mu}_\Delta$. It can be verified that if $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$, then $\bar{\boldsymbol{\mu}} \sim \boldsymbol{\mu}_\Delta$. This indeed confirms Lemma 2.1, and this is what the different transformations in Section 2.3 aim to achieve.

Let us now consider the case where the assumption $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$ (and thus $\bar{\boldsymbol{\mu}} \sim \boldsymbol{\mu}_\Delta$) is not (perfectly) satisfied. This happens for example if the transformations in Section 2.3 are computed based on approximations or estimates of the ground-truth statistics in the left column. In this case, we decompose $\bar{\boldsymbol{\mu}}$ into a component proportional to $\boldsymbol{\mu}_\Delta$ and a component orthogonal to $\boldsymbol{\mu}_\Delta$, i.e., $\bar{\boldsymbol{\mu}} = \delta\,\boldsymbol{\mu}_\Delta + \mathbf{e}$, with $\delta \in \mathbb{R}$ a scaling term and $\mathbf{e}$ an error vector orthogonal to $\boldsymbol{\mu}_\Delta$. $\mathbf{w}_{MILDA}$ can then be written as:

$$\mathbf{w}_{MILDA} \propto \hat{\Sigma}^{-1}\,((\phi + \delta)\,\boldsymbol{\mu}_\Delta + \mathbf{e})\,. \tag{29}$$

Depending on the statistics of the data, the same error does not always affect the projection vector $\mathbf{w}$ equally. The error will have a small effect on $\mathbf{w}_{MILDA}$ when the angle between $\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta$ and $\hat{\Sigma}^{-1}\,\mathbf{e}$ is small and/or when $\frac{1}{|\phi+\delta|}\frac{\|\mathbf{e}\|_2}{\|\,\boldsymbol{\mu}_\Delta\,\|_2}$ is small. By using the definition of ϕ in (28), we can further expand the latter expression as:

$$
\begin{aligned}
&\frac{1}{|\phi+\delta|}\frac{\|\mathbf{e}\|_2}{\|\,\boldsymbol{\mu}_\Delta\,\|_2} \\
&= \frac{1}{\left|1-\frac{q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\hat{\Sigma}^{-1}(\boldsymbol{\mu}_\Delta+\frac{\mathbf{e}}{\delta})}{1+q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\hat{\Sigma}^{-1}\boldsymbol{\mu}_\Delta}\right|}\frac{\|\mathbf{e}\|_2}{\delta\|\,\boldsymbol{\mu}_\Delta\,\|_2} \\
&= \left|\frac{1+q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\hat{\Sigma}^{-1}\boldsymbol{\mu}_\Delta}{1-q(1-q)\,\boldsymbol{\mu}_\Delta{}^\top\hat{\Sigma}^{-1}\frac{\mathbf{e}}{\delta}}\right|\frac{\|\mathbf{e}\|_2}{\|\,\bar{\boldsymbol{\mu}}-\mathbf{e}\|_2}. \qquad (30)
\end{aligned}
$$

Using (29) and (30), we can conclude that the following properties reduce the sensitivity of MILDA to errors in the ground-truth statistics:

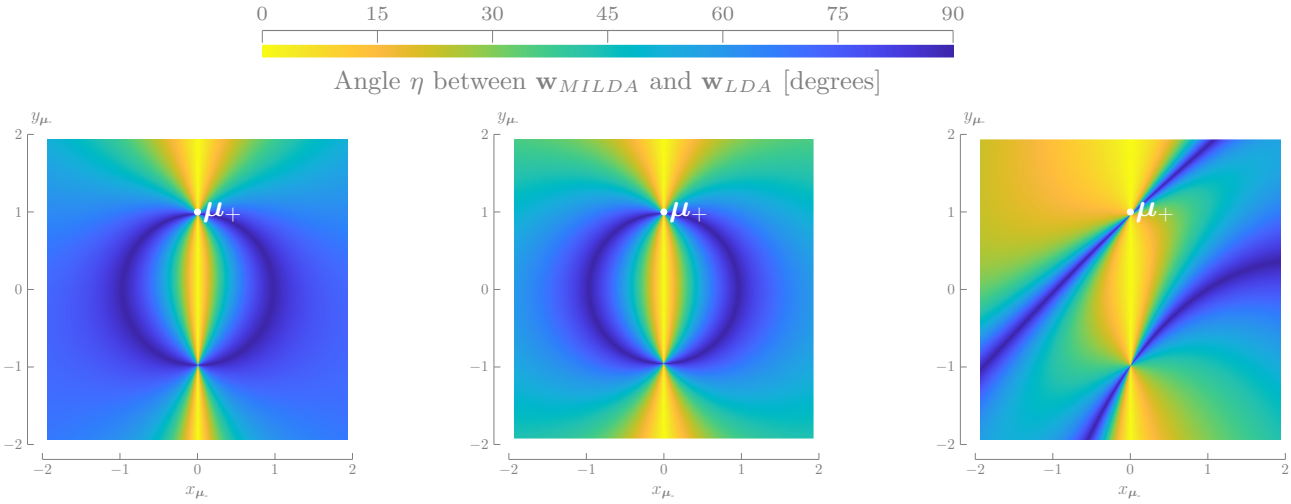
- Property 1: $\hat{\Sigma}^{-1}\,\mathbf{e}$ and $\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta$ form a small angle. Since $\mathbf{e}$ and $\boldsymbol{\mu}_\Delta$ are orthogonal (by construction), more skewed class covariances can reduce the impact of the error (although this is not guaranteed). Note that the third and fourth transformations of Table 1 first transform the data such that $\hat{\Sigma}=I$. For these transformations, $\hat{\Sigma}^{-1}\,\boldsymbol{\mu}_\Delta$ will thus always be orthogonal to $\hat{\Sigma}^{-1}\,\mathbf{e}$.
- Property 2: $\|\mathbf{e}\|_2$ is small relative to $\|\,\bar{\boldsymbol{\mu}}\,\|_2$. This is trivial: the error matters less when it is smaller. The worst case is when $\boldsymbol{\mu}_\Delta \perp \bar{\boldsymbol{\mu}}$, since in this case $\bar{\boldsymbol{\mu}}=\mathbf{e}$, such that $\mathbf{w}_{MILDA}\perp\mathbf{w}_{LDA}$ if $\hat{\Sigma}=I$. From (15), we see that $\boldsymbol{\mu}_\Delta{}^\top\,\bar{\boldsymbol{\mu}} = q\|\,\boldsymbol{\mu}_+\,\|_2^2 + (1-2q)\,\boldsymbol{\mu}_+{}^\top\,\boldsymbol{\mu}_- - (1-q)\|\,\boldsymbol{\mu}_-\,\|_2^2$. Since $\boldsymbol{\mu}_\Delta{}^\top\,\bar{\boldsymbol{\mu}}=0$ if $\boldsymbol{\mu}_\Delta\perp\bar{\boldsymbol{\mu}}$, this shows that this worst-case scenario does not occur when the class average of the class with the most samples is much larger than the other class average. If the classes are approximately balanced, it suffices that the norm of any one of the two class averages is much larger than the other.
- Property 3: There is a large class imbalance. Indeed, if $q\approx 0$ or $q\approx 1$, the first term in (30) becomes minimal (i.e. close to 1), therefore reducing the sensitivity to errors.
- Property 4: $\boldsymbol{\mu}_\Delta{}^\top\,\Sigma^{-1}\,\boldsymbol{\mu}_\Delta$ is small. This also minimises the first term in (30). Interestingly, this implies that the error will have a reduced effect on $\mathbf{w}_{MILDA}$ when there is a lot of overlap between the two classes. MILDA has thus the nice property that approximation errors matter less when the classes are harder to separate, which are precisely the cases where fully unsupervised methods do not perform well.

Beware that, in some cases, the beneficial effect of the latter two properties can be counteracted by other detrimental effects. This happens in the following cases.

- A large class imbalance will increase the error after transformation 1 from Table 1 (1 class average known) if the unknown class average is also the class with the least samples. Indeed, after the transformation the class with the most samples will be close to zero-mean resulting in a small $\bar{\boldsymbol{\mu}}'$, which directly counteracts the second aforementioned property.
- Both a large class imbalance and a worse separability make it harder to correctly estimate $\mathbf{d}$ when using transformations 3 or 4 from Table 1 (the covariance methods), since both will reduce the impact of the rank-1 update on the overall covariance matrix in (25).

3.2. Experimental sensitivity analysis

To visualise the findings from Section 3.1, the angle between the LDA and MILDA projection vectors is estimated in a two-dimensional space. To achieve this, we keep $\boldsymbol{\mu}_+$ fixed at $[0,1]^\top$. Meanwhile, $\boldsymbol{\mu}_- = [x,y]^\top$ is varied in the ranges $x,y\in[-2,2]$. For each $\boldsymbol{\mu}_-$, we compute the projection vectors $\mathbf{w}_{MILDA}\propto\bar{\Sigma}^{-1}\,\bar{\boldsymbol{\mu}}$ and $\mathbf{w}_{LDA}\propto(\Sigma_+ + \Sigma_-)^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$, assuming balanced classes. We then measure the similarity between the two projections using the angle η between $\mathbf{w}_{MILDA}$ and $\mathbf{w}_{LDA}$ using the formula $\cos(\eta)=\frac{\mathbf{w}_{LDA}^\top\,\mathbf{w}_{MILDA}}{\|\,\mathbf{w}_{LDA}\,\|_2\|\,\mathbf{w}_{MILDA}\,\|_2}$.



(a) $\hat{\Sigma} = I$ **(b)** $\hat{\Sigma} = 5I$ **(c)** $\hat{\Sigma} = 0.3I + 0.7J$

Figure 1: Heatmap representing the angle between $\mathbf{w}_{LDA}$ and $\mathbf{w}_{MILDA}$ as a function of the location of $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$. $\boldsymbol{\mu}_+$ is fixed at $[0, 1]^\top$, while $\boldsymbol{\mu}_-$ varies between $[-2, -2]^\top$ and $[2, 2]^\top$. The angle between the LDA and MILDA projection vector decreases when $\boldsymbol{\mu}_- << \boldsymbol{\mu}_+$ or $\boldsymbol{\mu}_- >> \boldsymbol{\mu}_+$, when the covariance increases, and when it is less white.

The closer η is to 0, the more the directions of $\mathbf{w}_{LDA}$ and $\mathbf{w}_{MILDA}$ are similar, and thus the more their respective projections are equivalent.

This set-up allows us to simulate a wide array of possible violations of the assumption $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$ up to an irrelevant scaling and rotation[5]. To study how the class covariance influences the sensitivity of MILDA to estimation errors, we will repeat this process for three different sets of class covariances. In the first experiment, $\hat{\Sigma}_1 = I$. In the second experiment, $\hat{\Sigma}_2 = 5I$. Here, the variance is larger, making the classes less separable. In the third experiment, $\hat{\Sigma}_3 = 0.3I + 0.7J$, with I the identity matrix and J a matrix with all ones. In this case, the class covariances are very skewed, allowing us to verify that this indeed decreases the influence of the error in many cases. For each experiment, $\Sigma_+ = \Sigma_- = \hat{\Sigma}$ and $q = 0.5$. To avoid any noise blurring the visualisation, no data was generated in this process. Instead, both the LDA and the MILDA projections were directly computed from $\Sigma_\pm$ and $\boldsymbol{\mu}_\pm$.

Figure 1 confirms the insights obtained from the mathematical analysis in Section 3.1. When $\hat{\Sigma} \propto I$ (Fig. 1a), the blue circle at the positions where $\| \boldsymbol{\mu}_+ \| \approx \| \boldsymbol{\mu}_- \|$ clearly shows the worst-case scenario where $\mathbf{w}_{MILDA} \perp \mathbf{w}_{LDA}$ in the area where $\bar{\boldsymbol{\mu}} = \mathbf{e}$ (see property 2). The region where $\boldsymbol{\mu}_+ \approx -\boldsymbol{\mu}_-$ is particularly interesting, as here the alignment assumption $\boldsymbol{\mu}_+ \sim \boldsymbol{\mu}_-$ is quite well satisfied, yet the sensitivity of MILDA to small deviations from such perfect alignment is very high because $\bar{\boldsymbol{\mu}} \approx \mathbf{0}$.

The ideal location for $\boldsymbol{\mu}_-$ to be, is either around $[0, 0]^\top$ (if $\boldsymbol{\mu}_+$ is non-zero) or $\pm\kappa\,\boldsymbol{\mu}_+$ with $\kappa \gg 1$ In these regions, the misalignment error between $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$ can be larger without having much effect on the solution.

Finally, we also clearly observe that MILDA becomes more robust against errors as the class covariances increase (Figure 1b) or become less white (Figure 1c), as predicted in Property 1 and Property 4 in the mathematical error analysis. This is interesting, as these are the typical scenario's where unsupervised methods suffer the most.

4. Ablation study

In this section, we demonstrate that MILDA closely matches the performance of a fully supervised LDA. The two models are compared to each other on a vast set of simulated data with varying parameters. This

[5]Scaling and rotation are irrelevant because LDA and MILDA transformations are affine invariant (proof omitted).

allows us to accurately investigate the influence of each parameter on the model's performance.

Note that the LDA and MILDA models would not be used interchangeably. LDA requires all statistics to be known, whereas MILDA is specifically designed for cases where much less information is available. Therefore, LDA should be seen as a benchmark that MILDA should ideally approach as closely as possible.

4.1. Experiment design

The models are validated on a simulated dataset containing N samples drawn from two different (possibly non-Gaussian) D-dimensional distributions, of which the properties are manipulated based on several hyperparameters. The samples are generated as $\boldsymbol{\mu}_{+/-} + \mathbf{n}_{+/-}(t) \in \mathbb{R}^D$ where the first term describes the class average and $\mathbf{n}_{+/-}(t)$ is a zero-mean stochastic vector which defines the shape of the distribution of the positive/negative class. The class mean of the positive class is defined as $\boldsymbol{\mu}_+ = [1, 0, ..., 0]^T \in \mathbb{R}^D$ (the one of the negative class will be case-specific and will be defined later on). The vectors $\mathbf{n}_{+/-}(t)$ are drawn from a distribution that is defined as a linear combination of a Gaussian $n_g \sim \mathcal{N}(\mathbf{0}, \Sigma_{+/-})$ and a Student's t distribution $n_t \sim \mathcal{T}(\mathbf{0}, \Sigma_{+/-}, \nu = 5)$ such that $n(t) = \frac{\gamma}{\sqrt{(1-\gamma)^2+\gamma^2}} n_g(t) + \frac{1-\gamma}{\sqrt{(1-\gamma)^2+\gamma^2}} n_t(t)$, $\gamma \in [0, 1]$. The covariance matrix of the positive class is defined as $\Sigma_+ = (1-\rho)I + \rho J$, with $I \in \mathbb{R}^{D\times D}$ the identity matrix and $J \in \mathbb{R}^{D\times D}$ a matrix of all ones. The covariance matrix of the negative class is defined as $\Sigma_- = (1-\sigma)\Sigma_+ + \sigma\Sigma_{orth}$, where $\Sigma_{orth} = VP\Lambda P^{-1}V^{-1}$, with V, Λ respectively the eigenvalues and eigenvectors of Σ_+ and P a random permutation matrix with only zeros on the diagonal. Σ_{orth} represents a distribution with the same covariance structure as Σ_+, but rotated such that the maximal covariance directions are orthogonal. The parameter σ can then be used to manipulate how similar Σ_+ and Σ_- are. The default hyperparameters are: $N = 1000$, $\boldsymbol{\mu}_- = [-1, \mathbf{0}_{D-1}]^\top$, $\rho = 0.2$ (correlated features), $D = 10$, $\gamma = 1$ (Gaussian distribution) , $\sigma = 0$ (i.e., Σ_+=Σ_-) and the class imbalance is set to $q = 0.3$, reflecting the fraction of samples from the positive class $\mathcal{C}_+$.

Each time, one of these default parameters is varied to study its influence on the behaviour of MILDA. The following parameters are studied:

- The effect of the class separation: $\boldsymbol{\mu}_- = [i, \mathbf{0}_{D-1}]^\top, i \in [-3, 1]$ (Fig. 2).
- The effect of the correlation between the features $\rho \in [0, 1]$ (Fig. 3).
- The degree to which Σ_+ and Σ_- are similar: $\sigma \in [0, 1]$ (Fig. 4).
- The data dimensionality $D \in [2, 100]$ (Fig. 5).
- The influence of the class imbalance $q \in [0.01, 0.99]$ (Fig. 6).
- The influence of the gaussianity of the class distributions, modelled by the fraction of Gaussian versus Student's t distribution $\gamma \in [0, 1]$ (Fig. 7).

Given that MILDA is a label-free method, we also compare it to two other popular unsupervised classification methods: K-means and Gaussian mixture models (GMMs). Where possible, these models were modified[6] to incorporate the same prior knowledge as MILDA to make the comparison as fair as possible. This is achieved by updating specific parameters at each iteration, following the updating procedure from Table 2. All other design choices for K-means and GMM are identical to the default Matlab R2021a implementation. The model is also compared to the unsupervised method that finds the projection vector that maximises skewness [20, 11], as it is a state-of-the-art unsupervised model that approximates the LDA projection vector. This establishes a baseline that MILDA must surpass to be considered useful. While LDA is the only true benchmark here (as this is MILDA's target performance), these additional comparisons are added to provide some insights into the strengths and limitations of MILDA as a label-free classifier compared to traditional unsupervised methods.

[6]For all experiments in this paper, we have verified that these modifications do not deteriorate the results compared to the original K-means and GMM algorithm.

The LDA model is always trained on the ground-truth statistics that are used to generate the samples, making it the *ideal* LDA model. Similarly, MILDA always uses the known ground-truth statistics to compute the transformations from Table 1. This reflects real-life limitations, as sample statistics of test data are generally unknown. Each experiment is repeated 50 times for the three ground-truth statistics (one class average known, the relative difference between class averages known and the class covariances known). The results are then averaged and shown in Figures 2 to 6. The shaded areas represent the standard deviation for each model. The performances of LDA and the skewness method are identical for all three ground-truth statistics since LDA always knows all ground-truth statistics and the skewness method has no access to any of the statistics.

4.2. Effect of class separation

Unsurprisingly, Figure 2 shows that a better class separation improves the performance of all models. MILDA performs significantly better than K-means, GMM and the skewness method when the class separation is large enough (e.g., when $\| \boldsymbol{\mu}_+ - \boldsymbol{\mu}_- \|_2 > 0.1$). MILDA is almost identical to LDA (at most 1% difference) in almost all settings, except for the case where only the class covariances are known (cases 3/4) if the class separation is small. In that case, the MILDA model is very sensitive to disparities between the ground-truth covariances and the estimated sample covariances. However, the other models never outperform MILDA, as they suffer even more from such data with poorly separated classes.

4.3. Effect of correlation between features

From Figure 3, we again see that MILDA and LDA achieve very similar performances, no matter how correlated the features are. If features are more correlated, the classes tend to become more separable, such that all methods except K-means improve in performance. K-means drops in performance because it implicitly assumes orthogonal class covariance matrices, i.e. uncorrelated features.

4.4. Effect of (dis)similarity between covariance matrices

The degree to which Σ_- and Σ_+ are different has no influence on either MILDA or LDA, as shown in Figure 4. MILDA thus reaches similar performances as LDA, no matter how different the class covariances are. The skewness method assumes homoscedacity, which explains its drop in performance as Σ_+ differs more from Σ_-. The GMM method outperforms LDA and MILDA when $\boldsymbol{\mu}_+$ is known and the two covariances are sufficiently orthogonal. This is because LDA no longer finds the optimal projection to separate two classes with different class covariances. Note that LDA and MILDA no longer optimise the same objective function, as $q \neq 0.5$ and $\Sigma_+ \not\propto \Sigma_-$. Figure 4 thus confirms that the difference in objective function does not lead to a drop in performance between MILDA and LDA.

4.5. Effect of data dimensionality

While the performance of K-means, GMM and the skewness method quickly drops with an increasing data dimensionality in Figure 5, MILDA only slowly loses performance. The slight drop in performance for MILDA compared to LDA is caused by the accumulation of small disparities between the ground-truth and sample statistics over an increasing data dimensionality. Note that LDA has access to all ground-truth statistics and thus doesn't suffer from these estimation errors. However, in reality, its statistics are often estimated on labelled data. In this case, LDA will also suffer from the curse of dimensionality.

Known	**K-means update**	**GMM update**
1) Class average $\boldsymbol{\mu}_+$	$\boldsymbol{\mu}_+^{(i)} \longleftarrow (1-\lambda)\boldsymbol{\mu}_+^{(i)} + \lambda\,\boldsymbol{\mu}_+$	$\boldsymbol{\mu}_+^{(i)} \longleftarrow (1-\lambda)\boldsymbol{\mu}_+^{(i)} + \lambda\,\boldsymbol{\mu}_+$
2) $\mathbf{d} \propto \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$	$\boldsymbol{\mu}_{+/-}^{(i)} \longleftarrow \left((1-\lambda)I + \lambda\mathbf{d}\mathbf{d}^\top\right)(\boldsymbol{\mu}_{+/-}^{(i)} - \bar{\boldsymbol{\mu}})$	$\boldsymbol{\mu}_{+/-}^{(i)} \longleftarrow \left((1-\lambda)I + \lambda\mathbf{d}\mathbf{d}^\top\right)(\boldsymbol{\mu}_{+/-}^{(i)} - \bar{\boldsymbol{\mu}})$
3) Σ_+, Σ_- and q or 4) $\Sigma_+ \propto \Sigma_-$	$\boldsymbol{\mu}_{+/-}^{(i)} \longleftarrow \boldsymbol{\mu}_{+/-}^{(i)}$	$\Sigma_{+/-}^{(i)} = (1-\lambda)\Sigma_{+/-}^{(i)} + \lambda\Sigma_{+/-}$

Table 2: The updating schemes for K-means and GMM to incorporate known pieces of information as a prior, with $\lambda \in [0, 1]$ and $\|\mathbf{d}\|_2 = 1$. By default, $\lambda = 0.7$ is chosen, as it empirically produced the best results.

4.6. Effect of class imbalance

Even at large class imbalances, MILDA performs equally well as LDA, as shown in Figure 6. They are the most different when only 1 class average is known (case 1) and when there are very few samples from the other class, and when only the covariances are known (case 3). This is expected, as in such cases, MILDA knows hardly anything about the minority class, making it difficult to properly estimate the projection vector, as explained in Section 3. A similar effect can be seen in the K-means, GMM and skewness method. The skewness method also drops in performance when both classes are balanced, as the method assumes requires a sufficiently large class imbalance in order to work.

4.7. Effect of (non-)Gaussianity

Even when both classes are fully sampled from a Student's t distribution with $\nu = 5$ degrees of freedom, which is known to have a lot of outliers, both LDA and MILDA achieve good results. GMM and the skewness method significantly drop in performance when the Gaussianity of the distribution decreases. While the GMM methods assumes Gaussianity, the skewness method does not. Instead, it drops in performance due to the many outliers introduced by the Student's t distribution.

4.8. Computational cost

Since MILDA relies on the same fundamental operations as standard LDA—namely, computing sample means, covariances, and performing an inverse operation—the asymptotic computational complexity remains $O(ND^2)+O(D^3)$, as in standard LDA. Thus, MILDA does not introduce additional computational overhead beyond what is already required for LDA.

Table 3 shows the average time (in ms) each model required to classify the samples using all default parameters. MILDA is only slightly slower than LDA, but this is mainly because LDA has access to the ground-truth covariance matrices. If LDA also needs to estimate Σ_+ and Σ_- from the data, there is no longer a difference in computation time. The K-means and GMM models take over 10 times longer to compute in this specific experiment. The skewness-based method takes 500 times longer to compute, even when the fast algorithm proposed in [11] is used.

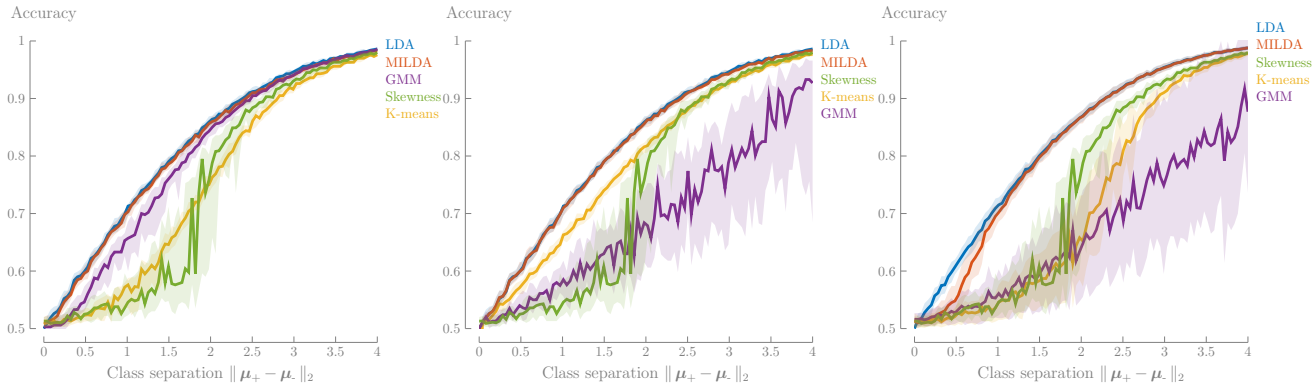
Prior knowledge	LDA	MILDA	K-means	GMM	Skewness
$\boldsymbol{\mu}_+$	0.24	0.31	3.0	4.7	143
$\mathbf{d}$	0.24	0.44	5.6	4.7	143
$\Sigma_{+/-}$	0.24	0.46	5.7	4.8	143

Table 3: Time required to classify all samples using default parameters [ms].

5. Illustration in signal detection use cases

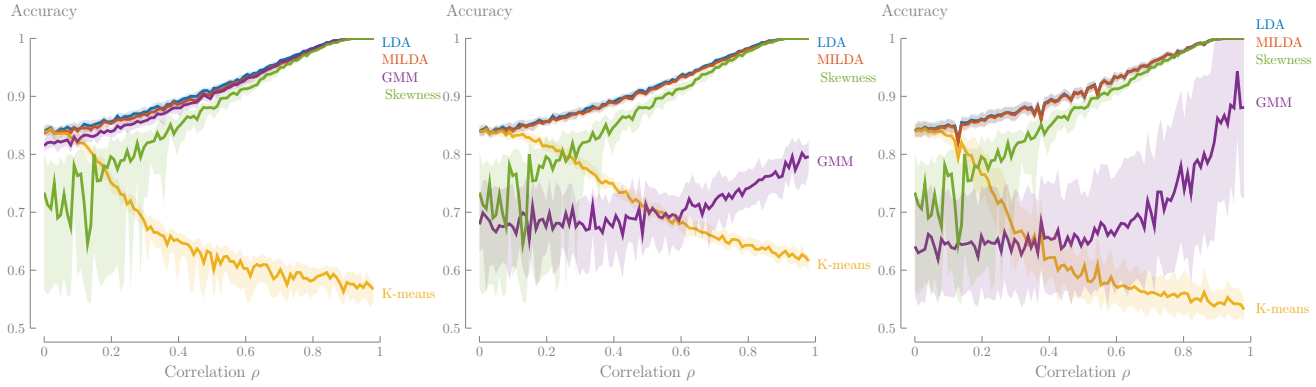
In this section, we provide some illustrative use cases where only a single ground-truth statistic is available, which form ideal use cases for MILDA. Note that this section is merely for illustrative purposes, where we only focus on simple examples for clarity.

First, we demonstrate how each type of ground-truth statistic can be estimated in a generic stationary signal detection problem. Then we demonstrate the utility of MILDA in a non-stationary detection problem. Finally, we apply MILDA on two real datasets: one where the ground-truth statistic is known and one where the statistic is roughly estimated on a few labelled samples in a semi-supervised context. Note that we apply MILDA here as a generic method for illustrative purposes, without claiming to outperform more advanced or tailored methods for these respective settings.



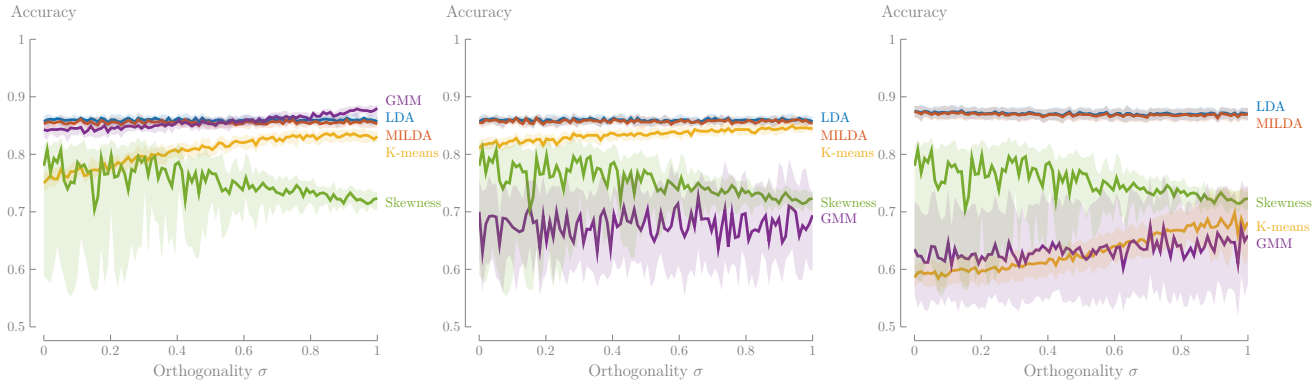
(a) μ_+ known **(b)** **d** known **(c)** Σ_+, Σ_-, q known

Figure 2: Accuracy as a function of class separation.



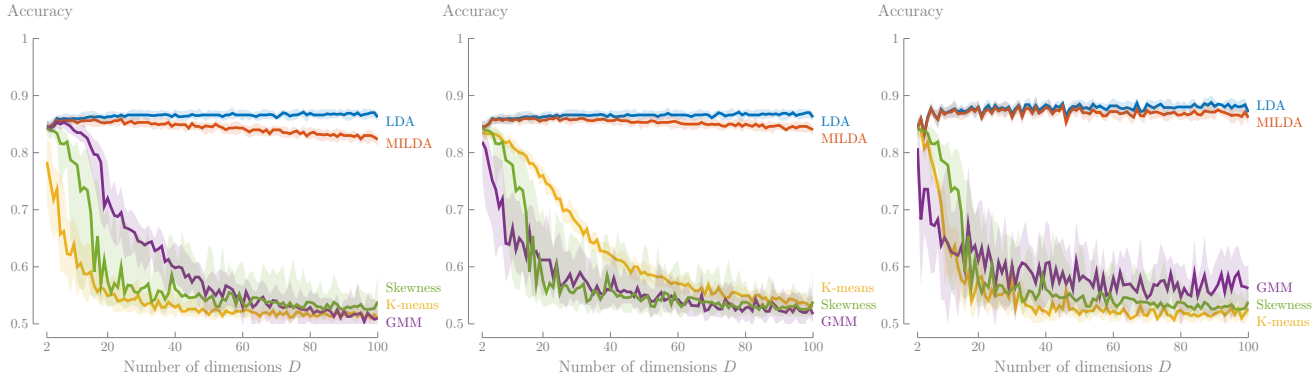
(a) μ_+ known **(b)** **d** known **(c)** Σ_+, Σ_-, q known

Figure 3: Accuracy as a function of the correlation between features.



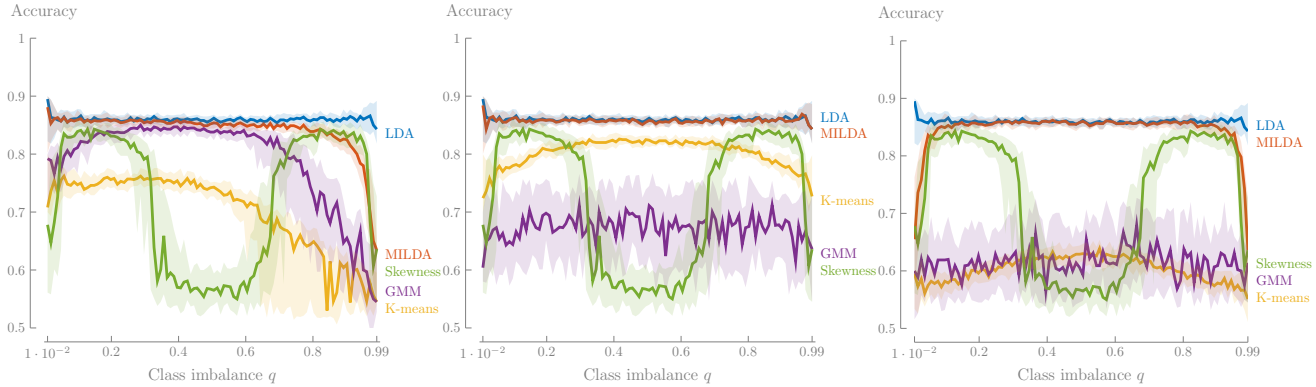
(a) μ_+ known **(b)** **d** known **(c)** Σ_+, Σ_-, q known

Figure 4: Accuracy as a function of how similar the class covariances are.



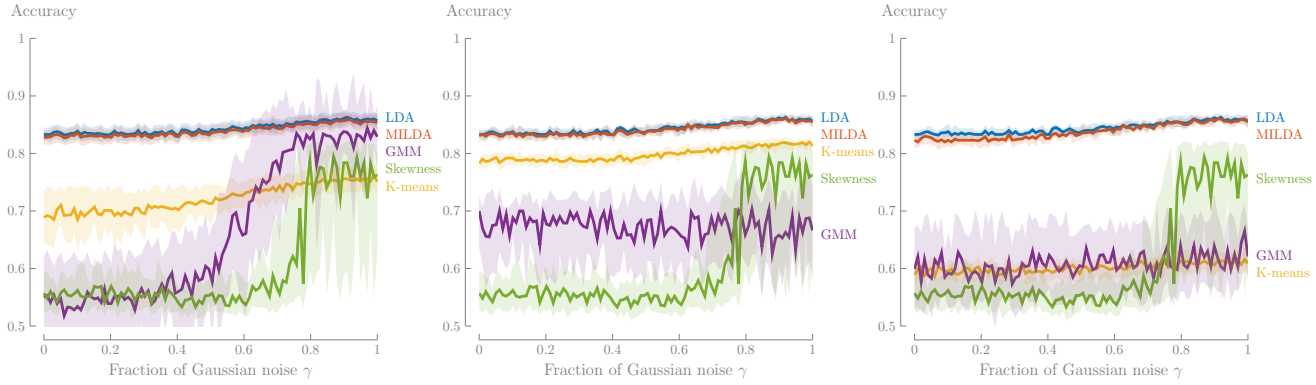
(a) $\boldsymbol{\mu}_+$ known **(b)** **d** known **(c)** Σ_+, Σ_-, q known

Figure 5: Accuracy as a function of the data dimensionality.



(a) $\boldsymbol{\mu}_+$ known **(b)** **d** known **(c)** Σ_+, Σ_-, q known

Figure 6: Accuracy as a function of the class imbalance.



(a) $\boldsymbol{\mu}_+$ known **(b)** **d** known **(c)** Σ_+, Σ_-, q known

Figure 7: Accuracy as a function of how Gaussian the distributions are.

5.1. Target signal detection

5.1.1. Experiment design

We consider a target signal detection problem in a D-channel sensor array. Given a target signal $s(t)$, the observed signal $\mathbf{x}(t) \in \mathbb{R}^{D\times 1}$ can be modelled as $\mathbf{x}(t) = s(t)\mathbf{a} + \mathbf{n}(t)$, with $\mathbf{a} \in \mathbb{R}^{D\times 1}$ the steering vector and $\mathbf{n}(t) \in \mathbb{R}^{D\times 1}$ the noise. All signals are assumed to be stationary. To illustrate the different ways that MILDA can be used, we propose three sub-problems:

- In the first problem, we attempt to detect whether or not the target signal is present, i.e. $s(t) \in \{0, 1\}$. The steering vector $\mathbf{a}$ is unknown, but assumed fixed within the observation window that contains the N samples that have to be classified. Even though the general noise statistics are unknown, the sensors are calibrated such that the noise is zero-mean. One class average ($\boldsymbol{\mu}_{noise} = \mathbf{0}$) is thus known, allowing us to use MILDA with transformation 1 from Table 1.
- In the second problem, a binary signal $s(t) \in \{-1/2, 1/2\}$ is sent from a single location. In this case, the signal $s(t)$ is always active (i.e., it is never zero). The steering vector $\mathbf{a}$ is known (up to an arbitrary scaling) [21]. The statistics of $s(t)$ and $\mathbf{n}(t)$ are assumed to be unknown. In this case, we know that the relative difference between both classes averages is proportional to the steering vector ($\boldsymbol{\mu}_+ - \boldsymbol{\mu}_- \propto \mathbf{a}$). We can therefore use MILDA with transformation 2 from Table 1.
- In the third problem, two different locations interchangeably transmit the signal $s(t) = 1$ while the other location is silent. For each observation $\mathbf{x}(t)$, we aim to detect which location transmitted the signal. The locations (and corresponding steering vectors) are unknown, similar to case 1. The noise covariance matrix $S = \mathbb{E}[\mathbf{n}(t)\mathbf{n}^\top(t)]$ is assumed to be known. It is therefore possible to use MILDA with transformation 4 from Table 1.

We set $D = 10$ and perform multiple Monte-Carlo runs. In each run, $N = 1000$ samples are created with equal probability between the two detection classes of the corresponding experiment, i.e., $N/2$ of the samples belong to class 0, the other $N/2$ belong to class 1. The noise is modelled as $\mathbf{n}(t) = \mathbf{g}(t) + \mathbf{f}(t)$, with Gaussian noise $\mathbf{g}(t) \sim \mathcal{N}(0.3\,\mathrm{diag}(\boldsymbol{\sigma}) + 0.7J)$ and a high-frequency interference signal $\mathbf{f}(t) = \boldsymbol{\alpha} \odot \sin(\pi/2t + \boldsymbol{\phi})$. The variances $\boldsymbol{\sigma} \in \mathbb{R}^{D\times 1}$ and amplitudes $\boldsymbol{\alpha} \in \mathbb{R}^{D\times 1}$ are randomly sampled from the uniform distribution $\mathcal{U}_D(0, 1)$, the phase of the interferer $\boldsymbol{\phi} \in \mathbb{R}^{D\times 1}$ is sampled from $\mathcal{U}_D(0, 2\pi)$, and the steering vectors are sampled from $\mathcal{U}_D(-1, 1)$ (in all cases the D different dimensions are sampled independently from these distributions). Each experiment is repeated 100 times. All models are trained in the same way as in Section 4. Since LDA has access to information that wouldn't be available in the described scenarios, it merely serves as a benchmark.

5.1.2. Results

The detection accuracy for each model and each experiment is displayed in Table 4. Similar to the previous experiments, LDA and MILDA achieve comparable performances, despite the fact that the former uses more knowledge. The inclusion of non-Gaussian noise further demonstrates that MILDA does not require the assumption of Gaussianity to achieve a performance comparable to LDA.

Problem	LDA	MILDA
Zero-mean noise	93.4 (4.4)	92.8 (4.4)
Binary signal	83.2 (4.4)	82.9 (4.4)
Noise covariance	90.3 (4.6)	90.2 (4.6)

Table 4: The average accuracy (%) and standard deviation between brackets for each model. MILDA performs only slightly worse than LDA, even though it requires significantly less information.

5.2. Non-stationary data

5.2.1. Experiment design

Until now, all experiments were performed on stationary data. However, there is a vast amount of classification problems where the stationarity assumption does not hold. Instead, the class distributions can shift and morph over time. Ideally, the classification models should adapt to these changes. While this is generally possible for MILDA (assuming the ground-truth statistic is stationary), this is rarely the case for LDA due to the absence of labels to re-estimate the relevant statistics at inference time to recompute the LDA projection vector.

To build further on the first problem from Section 5.1, we illustrate the effects of non-stationary data on binary target signal detection in zero-mean noise by introducing three changes in the data and tracking the accuracy over time (consisting of 10,000 samples in total):

- The first change is a sudden change in the noise covariance from Σ_1 to Σ_2 at sample 2,500.
- The second change is a *gradual* drift in both the steering vector from $\mathbf{a}_1$ to $\mathbf{a}_2$ and the noise covariance from Σ_2 to Σ_3 between samples 5,000 and 7,500 .
- The third change is a sudden change in the steering vector from $\mathbf{a}_2$ to $\mathbf{a}_3$ at sample 7,500.

Note that while the statistics of the noise change over time, the zero-mean assumption is assumed to remain valid over the entire experiment, which is exploited by MILDA. Similar to the first problem, $s(t) \in \{0, 1\}$ and $\mathbf{a}_i \sim \mathcal{U}_D(-1, 1)$. The noise is modelled in the same way as in Section 5.1. The signal and noise statistics are changed by respectively resampling $\mathbf{a}$ and $\boldsymbol{\sigma}$, $\boldsymbol{\alpha}$, $\boldsymbol{\phi}$. During the gradual change, each parameter is changed by linearly interpolating between the value at the start and at the end of the gradual change. The norm of the steering vector is also changed by linearly interpolating between $\|\mathbf{a}_1\|$ and $\|\mathbf{a}_2\|$. This guarantees that the expected class separation is constant during the gradual shift.

We performed 1000 Monte-Carlo runs. MILDA is adaptively retrained using a moving window covering the last 500 samples. LDA is trained using the original statistics Σ_1 and $\mathbf{a}_1$ and thus does not adapt to the changes in statistics (as there are no labels available to update the model). Since the noise is always zero-mean, MILD uses $\boldsymbol{\mu}_{noise} = \mathbf{0}$ as the stationary ground-truth statistic.

5.2.2. Results

The average accuracy of the four models is shown in Figure 8. As expected, LDA slightly outperforms MILDA on samples that are drawn from the same distribution as it was trained on. LDA and MILDA equally drop in performance when sudden changes are introduced in the statistics. However, MILDA completely recovers after 500 samples, which is the width of the moving window MILDA was trained on. Remarkably, MILDA barely drops in performance during the gradual change in statistics (on average only 0.25%), showing that the model is well able to track gradual changes in statistics. As expected, LDA continuously drops in performance as more statistics are changed. Since it is unable to adapt due to the lack of labels, it is also unable to recover from these changes.

5.3. Steady-state visually evoked potential classification

5.3.1. Experiment design

We apply the MILDA framework to a Brain Computer Interface (BCI) use case for the classification of steady-state visually evoked potentials (SSVEP) in electroencephalogram (EEG) data. In this problem, a subject is presented with a screen where multiple squares are flickering at different frequencies, while the subject's brain activity is measured with an EEG recording device.

When the subject focuses on one of these flickering squares, this generates a neural response in the brain that is phase-locked to this flickering pattern, and which is known as an SSVEP. The SSVEP is a periodic neural signal with a fundamental frequency that is equal to the flickering frequency. However, it is not possible to observe this neural response in the EEG with the naked eye and statistical methods are needed to detect its presence. By using squares with different flickering frequencies, each representing a different

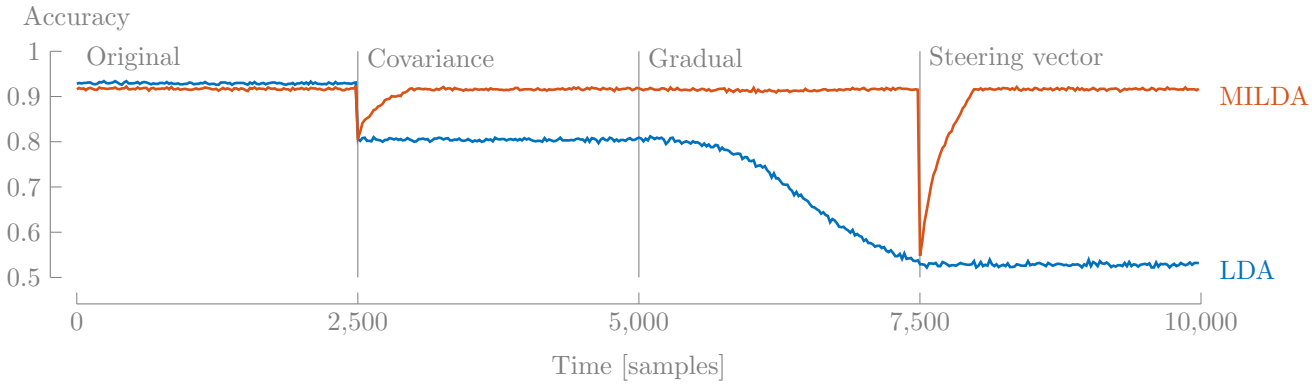


Figure 8: Comparison of LDA and MILDA on non-stationary data. Since MILDA does not require labels to train, it can adapt to both sudden and gradual changes in the statistics.

symbol, it is possible to design a brain-computer interface to communicate via brain signals. The goal is then to classify from the EEG which flickering pattern (i.e. symbol) the subject focuses on.

The SSVEP detection is typically done by fitting a model between the EEG and a parametric description of the SSVEP response. The features that are used to detect the SSVEP then consist of a set of metrics that quantify how well the model fits the EEG measurements for a given test window of a few seconds. While there exist various models and fitting procedures, we employed the popular CCA procedure as explained in [22]. We refer to Appendix A for a more in-depth explanation of the model, the feature construction, and implementation details. For the purpose of this paper however, it is sufficient to know that this method produces a D-dimensional feature vector for each test window, which contains a set of correlation coefficients between the observed EEG in the test window and D canonical components in the fitted model. In the absence of an SSVEP, there is no correlation between the EEG and the model, in which case the feature vector is on average zero. The latter is prior knowledge that can indeed be leveraged in the MILDA procedure.

This experiment uses a subset[7] of the publicly available SSVEP dataset BETA [23]. In this subset, 55 participants look at a stimulus flickering at one of five possible frequencies (8 Hz, 9 Hz,10 Hz,11 Hz and 12 Hz) for 3 s. This process is repeated a total of twenty times, with the participant looking exactly four times at each frequency. The EEG is recorded with $C = 64$ channels. For more details, see [23].

The CCA model is fitted using leave-one-subject-out (LOSO) cross-validation, such that the subject under test is excluded from the fitting procedure. Five different models, one for each target frequency, are fitted with $D = 10$ canonical components (see Appendix A). After fitting these models, we split the EEG data into N non-overlapping test windows with a length of 1 s each, and use each model to translate the test windows into 10-dimensional feature vectors that quantify the model (mis)match for the corresponding window. One such a 10-dimensional feature vector is constructed for each target frequency (i.e., five times) and each test window (N times).

These five feature vectors for each test window are then projected by a classifier onto five 1-dimensional scores. The classifiers aim to maximally differentiate between feature vectors where the flickering frequency matches the attended frequency and feature vectors where this is not the case. Afterwards, the frequency corresponding to the largest out of the five scores is selected as the predicted attended frequency for that test window. This is compared to the actual attended frequency to compute the final accuracy. Similar to Section 4, we use LDA, MILDA, K-means, GMM and the skewness method as classifiers. Moreover, since the features may not be linearly separable, we also compare with two popular non-linear dimensionality reduction techniques, namely t-distributed Stochastic Neighbor Embedding t-SNE [24] and non-linear PCA through an auto-encoder as proposed in [25].

[7]The original dataset contains 40 different frequencies and 70 subjects. 15 subjects were left out because the stimulus was presented for 2 s instead of 3 s.

Model	LDA	MILDA	K-means	GMM	Skew.	t-SNE	NLPCA
Classification accuracy	73.9	72.8	70.1	36.2	74.0	63.2	68.5
Standard deviation	± 18.3	± 19.0	± 16.7	± 12.6	± 18.2	± 18.8	± 17.5

Table 5: The average accuracy [%] and standard deviation to classify the attended frequency out of three possible frequencies correctly for LDA and MILDA in an SSVEP task. MILDA slightly outperforms LDA, although the difference is not significant.

The LDA projection vector is trained using leave-one-subject-out cross-validation and uses the ground-truth frequency labels of each test window. The MILDA, K-means and GMM algorithms only rely on the fact that the class average is **0** when the SSVEP for the target frequency is not present and uses all available data of all subjects to compute the group statistics. The skewness, t-SNE and NLPCA methods are fully unsupervised. The default *tsne* function in Matlab was used for t-SNE, while the nlpca toolbox from [25] was used for nonlinear PCA. We used default hyperparameters for both methods.

5.3.2. Results

As shown in Table 5, the unsupervised MILDA model closely approximates the performance of the supervised LDA model. The difference is not significant ($p = 0.23$, Wilcoxon signed-rank). The skewness method also achieves a similar performance to the supervised LDA method, although it is more than 100 times more expensive to compute than MILDA. The nonlinear TSNE and NLPCA models perform significantly worse than LDA and MILDA ($p < 0.001$), while being respectively 10^4 and 10^5 times more expensive. Although such nonlinear models can be much more potent than LDA for some applications, they are also more prone to overfitting on smaller datasets, such as this one. This performance trade-off between linear and non-linear models is heavily application dependent.

Surprisingly, all models are most likely to confuse between the 8 Hz and 12 Hz frequencies, followed by the pairs [8 Hz,10 Hz] and [10 Hz,12 Hz]. In fact, the pair [8 Hz, 12 Hz] is 20% more probable to be confused than any other pair. This is likely because these frequencies share some harmonics, respectively 24 Hz, 40 Hz and 60 Hz. This demonstrates the importance to avoid frequencies with common divisors as much as possible when performing SSVEP.

5.4. MILDA for semi-supervised learning: fraud detection use case

In an ever growing number of applications, a large amount of data is available, but it is too time consuming and too costly to manually label this data. This has motivated research in semi-supervised learning, where both a limited amount of labelled data and a large amount of unlabelled data are used to train a model. MILDA can be applied in such a semi-supervised framework by estimating one or both class averages using the limited available data and then estimating the MILDA projection vector on the unlabelled data. This approach can outperform a supervised LDA model trained on the labelled data because: 1) less data is typically required to accurately estimate a class averages compared to a covariances matrix and 2) MILDA is less sensitive to estimation errors, as explained in Section 3.

To demonstrate this, both MILDA and LDA were applied on a publically available credit card fraud detection dataset [26]. This dataset contains 284 807 transactions, of which only 492 (0.172%) of the transactions are fraudulous. The dataset contains 30 features for each transaction. For privacy concerns, the exact nature of these features is mostly unknown [26].

5.4.1. Experiment design

We consider three potential use cases. In the first case, no information is known about the fraudulous transactions. Instead, we use the features from approximately 1160 non-fraudulous transactions to estimate their class average. In the second case, only information about 2 fraudulous transactions are known. Therefore, these 2 fraudulous transactions are used to estimate their class average. In the third case, both class averages are used together to estimate the relative difference between class averages. In all three cases, these ground-truth statistics are then used to train MILDA as outlined in Algorithm 1. The LDA model is trained on the full labelled subset of 1160 non-fraudulous and 2 fraudulous transactions. All models are then

evaluated on all remaining validation data. This process is repeated 246 times, such that each transaction was used exactly once for supervised training. Since the 246 repetitions are not independent due to the large overlap in test data, only the average performance is reported. Similar to Section 5.3, the K-means and GMM models are trained on the same data as MILDA, while the skewness, t-SNE and NLPCA models are fully unsupervised.

Due to the large class imbalance, accuracy is an unreliable metric to measure performance. Instead, we measure the performance of each model using the area under the receiver operand curve (AUC). The chance level of the AUC is always 0.5, irrespective of the class imbalance, while the AUC of a perfect model is 1.

5.4.2. Results

As shown in Table 6, MILDA achieves an AUC of 91.1% when the ground-truth class average was computed with only two fraudulous transactions. The AUC drops to 82.1% when only labels from the larger non-fraudulous class are available, which is comparable to the AUC of the supervised LDA model. The best performing fully unsupervised method is t-SNE, which achieves an AUC of 68.3%. In contrast to MILDA, the adapted K-means and GMM algorithms were unable to efficiently leverage the two labelled samples, with an AUC of respectively 68.1% and 72.3%. The high performance of MILDA is also combined with a very low computational cost, as shown in Table 7. MILDA and LDA are at least one order of magnitude faster than all other models.

This demonstrates that MILDA is suprisingly capable to leverage the two labelled samples for an accurate classification of credit card fraud. This result is partially caused by the low sensitivity of MILDA to approximation errors when there is a large class-imbalance. In such scenarios, samples of the underrepresented class should ideally be used to estimate a class average, even when they are much less numerous. This is in line with previous observations in the ablation study of Section 4.

Labelled training data	LDA	MILDA	K-means	GMM	Skew.	t-SNE	NLPCA
Only non-fraudulous	-	82.1	69.4	61.0	64.1	68.3	53.6
Only fraudulous	-	91.1	62.2	63.7	64.1	68.3	53.6
Fraudulous and non-fraudulous	82.3	91.1	68.8	60.8	64.1	68.3	53.6

Table 6: The area under the ROC curve (AUC) [%] on a credit card fraud detection task where either 1160 non-fraudulous transactions, 2 fraudulous transactions or all 1162 transactions are used as labelled training data. LDA can only be trained when both classes are labelled. The skewness, t-SNE and NLPCA models are always fully unsupervised and are merely added for reference.

5.5. Multi-class classification with MILDA

The MILDA framework proposed in this paper can be extended to multi-class problems, albeit to a limited extend. This is possible with a one-versus-all approach provided that the class averages of all but one class are known. Given C known class averages, C projection vectors are computed using Algorithm 1. Each time, a different class average is subtracted from the feature vectors $\mathbf{x}_i$.The resulting projections have values close to zero when a feature is likely to belong to the class which average was subtracted, and values far from zero otherwise. If all projections of a feature vector are far from zero, then the feature could be classified as an outlier.

The usability of MILDA in multi-class scenarios is nevertheless more limited than in two-class scenarios. Knowledge of the relative difference between class averages up to a scaling or the covariance of each class are not sufficient to perform multi-class classification with MILDA. Furthermore, MILDA cannot be used in the often used one-versus-one approach for multi-class classification, since the lack of labels makes it impossible to isolate two classes from all other classes.

Labelled training data	LDA	MILDA	K-means	GMM	Skew.	t-SNE	NLPCA
Computation time [s]	0.35	0.17	2.52	41.2	59.9	2792.2	7335.2

Table 7: The average time [s] required to train and validate each model on the full credit card fraud detection dataset.

6. Conclusion

In this paper, we have proposed the minimally informed LDA (MILDA) model that can calculate the same projection vector as LDA without requiring labelled data, assuming prior knowledge of one of the following ground-truth statistics: a single class average, the relative difference between two class averages (up to a scaling) or the class covariance matrices (up to a scaling). We have mathematically proven that the MILDA projection vector is equivalent to the LDA projection vector under the mild assumption that the classes are balanced or that the class covariances have a similar shape. When both assumptions are violated, MILDA optimises a slightly different objective function, yet we could not find a noteworthy difference in performance between both. Furthermore, we have shown both mathematically and empirically that MILDA has the nice property of being robust in cases where the classes are harder to separate, features are more correlated or when the classes are more unbalanced, which are all cases where unsupervised methods typically have difficulties with. We validated MILDA on both simulated and real data. The extensive simulated experiments demonstrated the robustness of the proposed method in a wide range of conditions, covering both unsupervised and semi-supervised problems.

In contrast to most unsupervised LDA models, the MILDA projection vector can be computed in closed form with a computational cost comparable to the LDA projection vector. Because MILDA has a low computational cost to train and does not require any labels, it is well-suited for use in adaptive classifiers that update themselves over time using the most recent data.

Appendix A. Feature extraction for SSVEP decoding

The correlation features for SSVEP are computed with a canonical correlation analysis (CCA) model that extracts decoder-encoder pairs that maximise the correlation between the recorded EEG $\mathbf{x}(t) \in \mathbb{R}^{C\times 1}$ and the parametric description of the SSVEP response $\mathbf{y}(t,f) = [\mathbf{y}_1(t,f)\dots\mathbf{y}_N(t,f)]^\top$ with $\mathbf{y}_n(t,f) = [\sin(2\pi nft), \cos(2\pi nft)]$, C the number of channels and f the frequency of the attended flickering square. $n = 1\dots N$ enables the inclusion of higher harmonics in $\mathbf{y}(t)$ [22].

CCA computes a set of orthogonal decoders $\mathbf{d} \in \mathbb{R}^{C\times 1}$ and encoders $\mathbf{e} \in \mathbb{R}^{2N\times 1}$ that maximise the Pearson correlation between $\mathbf{d}^\top\mathbf{x}(t)$ and $\mathbf{e}^\top\mathbf{y}(t)$ based on training data for which f is known (details in [23]). The solution can be shown to be the solutions of the following joint generalised eigenvalue problem:

$$R_{xx}\mathbf{d} = \lambda R_{xy}R_{yy}^{-1}R_{xy}^\top\mathbf{d} \tag{A.1}$$

$$R_{yy}\mathbf{e} = \lambda R_{xy}^\top R_{xx}^{-1}R_{xy}\mathbf{e}, \tag{A.2}$$

with R_{xy} the cross-correlation matrix between $\mathbf{x}(t)$ and $\mathbf{y}(t,f)$ and R_{xx}, R_{yy} the autocorrelation matrix of $\mathbf{x}(t)$ and $\mathbf{y}(t,f)$ respectively.

The Pearson correlation between $\mathbf{d}^\top\mathbf{x}(t)$ and $\mathbf{e}^\top\mathbf{y}(t,f)$ on a test window then serves as a feature to detect whether the SSVEP is present or not at the frequency f. CCA provides up to $D = \min(C, 2N)$ such decoder-encoder pairs in order of decreasing correlations [22], which allows to build a feature vector containing up to D correlation coefficients. By construction, in the absence of an SSVEP at target frequency, the corresponding feature vector will be 0 in expectation as no correlation is present.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009, series Title: Springer Series in Statistics.

[2] C. Bouveyron, G. Celeux, T. B. Murpy, and A. E. Raftery, *Model-Based Clustering and Classification for Data Science*. Cambridge University Press, 2019.

[3] J. Liu, X. Wang, R. Wang, C. Xu, R. Zhao, H. Li, S. Zhang, and X. Yao, "Near-infrared auto-fluorescence spectroscopy combining with Fisher's linear discriminant analysis improves intraoperative real-time identification of normal parathyroid in thyroidectomy," *BMC Surgery*, vol. 20, no. 1, pp. 1–7, Jan. 2020.

[4] C. W. Antuvan and L. Masia, "An LDA-Based Approach for Real-Time Simultaneous Classification of Movements Using Surface Electromyography," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 27, no. 3, pp. 552–561, 2019.

[5] R. Fu, Y. Tian, T. Bao, Z. Meng, and P. Shi, "Improvement Motor Imagery EEG Classification Based on Regularized Linear Discriminant Analysis," *Journal of Medical Systems*, vol. 43, no. 6, 2019.
[6] C. Ricciardi, A. S. Valente, K. Edmund, V. Cantoni, R. Green, A. Fiorillo, I. Picone, S. Santini, and M. Cesarelli, "Linear discriminant analysis and principal component analysis to predict coronary artery disease," *Health Informatics Journal*, vol. 26, no. 3, pp. 2181–2192, 2020.
[7] D. Valuyskiy, S. Vityazev, and M. Vashkevich, "Roller Vibration Detection with Radar Sensor and Linear Discriminant Analysis," in *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, Mar. 2024.
[8] G. Devisetty and D. J. Rani, "Analysis and comparison for accurate prediction of innovative Bradycardia using linear discriminant analysis classifier and support vector machine," *AIP Conference Proceedings*, vol. 3193, no. 1, Nov. 2024.
[9] R. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936.
[10] D. Peña and F. J. Prieto, "The kurtosis coefficient and the linear discriminant function," *Statistics & Probability Letters*, vol. 49, no. 3, pp. 257–261, Sep. 2000.
[11] N. Loperfido, "Skewness-based projection pursuit: A computational approach," *Computational Statistics & Data Analysis*, vol. 120, pp. 42–57, Apr. 2018.
[12] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*. JSTOR, 1974.
[13] S. P. Lloyd, "Least Squares Quantization in PCM," in *IEEE Transactions on Information Theory*, vol. 28, 1982, pp. 129–137.
[14] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *ACM-SIAM Symposium on Discrete Algorithms*, 2007.
[15] F. De La Torre and T. Kanade, "Discriminative cluster analysis," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 241–248.
[16] P. Deng, H. Wang, T. Li, S. J. Horng, and X. Zhu, "Linear discriminant analysis guided by unsupervised ensemble learning," *Information Sciences*, vol. 480, pp. 211–221, Apr. 2019.
[17] Q. Wang, F. Wang, F. Ren, Z. Li, and F. Nie, "An Effective Clustering Optimization Method for Unsupervised Linear Discriminant Analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3444–3457, Apr. 2023.
[18] R. Martín-Clemente and V. Zarzoso, "LDA via L1-PCA of Whitened Data," *IEEE Transactions on Signal Processing*, vol. 68, p. 225, 2020.
[19] M. A. Woodbury, *Inverting modified matrices*. Princeton, NJ: Department of Statistics, Princeton University, 1950.
[20] N. Loperfido, "Skewness and the linear discriminant function," *Statistics & Probability Letters*, vol. 83, no. 1, pp. 93–99, Jan. 2013.
[21] S. Markovich, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 6, pp. 1071–1086, Aug. 2009.
[22] Z. Lin, C. Zhang, W. Wu, and X. Gao, "Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 6, pp. 1172–1176, Jun. 2007.
[23] B. Liu, X. Huang, Y. Wang, X. Chen, and X. Gao, "BETA: A Large Benchmark Database Toward SSVEP-BCI Application," *Frontiers in Neuroscience*, vol. 14, Jun. 2020.
[24] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.
[25] M. Scholz, "Validation of Nonlinear PCA," *Neural Processing Letters*, vol. 36, no. 1, pp. 21–30, Aug. 2012.
[26] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.