**Advanced Data Analysis – Part I**

Assignment – Feature Extraction and Evaluation

Université de Technologie de Compiègne

Ingénierie des Systèmes Complexes [M2]

Ajet KETA

Megi XHAFA

Contents

[**1.** **Introduction** 3](#_Toc147681762)

[**2.** **Loading Data** 3](#_Toc147681763)

[**3.** **Feature Extraction and Discriminative Power Evaluation** 4](#_Toc147681764)

[**3.1** **Statistical Features** 4](#_Toc147681765)

[**3.1.1** **Mean** 4](#_Toc147681766)

[**3.1.2** **Variance** 5](#_Toc147681767)

[**3.1.3** **Skewness** 6](#_Toc147681768)

[**3.1.4** **Kurtosis** 7](#_Toc147681769)

[**3.1.5** **Number of Window Variances Within General Variance Range** 8](#_Toc147681770)

[**3.2 Temporal Features** 9](#_Toc147681771)

[**3.2.1 Energy** 9](#_Toc147681772)

[**3.2.2 Root Mean Square** 10](#_Toc147681773)

[**3.3 Spectral Features** 11](#_Toc147681774)

[**3.3.1 Power** 12](#_Toc147681775)

[**3.3.2 Mean Power Frequency** 13](#_Toc147681776)

[**3.3.2 Moment Coefficient of Skewness** 13](#_Toc147681777)

[**3.3.3 Moment Coefficient of Kurtosis** 14](#_Toc147681778)

[**3.3.4 Energy by Frequency Bands** 15](#_Toc147681779)

[**3.3.4 High/Low Ratio** 16](#_Toc147681780)

[**4.** **Evaluation** 17](#_Toc147681781)

# **Introduction**

This practical session revolves around the fundamental objective of becoming well-accustomed in feature extraction procedures. These procedures are explored through the realm of signal processing and statistics. The foundation of this practical work lies in a simulated dataset consisting of raw surfaced Electromyography (EMG) data. This dataset mimics real-world scenarios with a five-second duration and a sampling frequency of 10kHz. The dataset is separated into two different classes (EMG1-EMG10 and EMG11-EMG20) which represent different levels of muscle contractions.

Within this context, a comprehensive number of features, including temporal metrics like energy and RMS value, spectral characteristics such as moments and energy within specific frequency bands, as well as statistical measures like *mean, variance, skewness,* and *kurtosis*, have been programmed in the MATLAB workspace.

Alongside the extraction of features, in this practical work we have gone through the evaluation of the discriminative power of each feature through a combination of boxplots, Area Under the Curve (AUC), and accuracy metrics computed from Receiver Operating Characteristic (ROC) curves.

In essence, this assignment goes through the practical aspects of feature extraction, providing insights into the strengths and weaknesses of different features when applied to EMG data, and ultimately enabling informed decision-making for future machine learning tasks.

# **Loading Data**

The first task of this assignment consists of merging the data from EMG1 to EMG20 MATLAB files provided and loading them in the workspace. This will result in a 20x50000 matrix where for each of the 20 rows, we have 50000 samples. Furthermore, we must separate the data from this initial matrix into two classes, class one being EMG1 to EMG10 and class two being EMG11 to EMG20, each representing different muscle contraction levels. The dimensions of the matrixes for these classes will be 10x50000. Throughout the development of the code however, we have only considered the initial 20x50000 matrix and we’ve done the class separation in the index specification.

A screenshot of a computer

Description automatically generated

Figure Loading Data

# **Feature Extraction and Discriminative Power Evaluation**

During this phase of the practical work, we go through the feature extraction process to obtain temporal, spectral and statistical features for this dataset. For each considered feature we have computed the discriminative power using Boxplot, Receiver Operating Characteristics (ROC) curve, Area Under the Curve (AUC), Accuracy, Specificity and Sensitivity. These tools are chosen to do the evaluation for each feature based on their important characteristics and descriptive abilities.

Boxplot is a graphical representation of the distribution of a dataset. It displays key summary statistics, including the median, quartiles, and potential outliers. The box represents the interquartile range, which spans from the first quartile to the third, while the line represents the median. The whiskers extended from the box reach the minimum and maximum values. Using the boxplot representation, we can also define the outliers which fall above or below the whiskers.

The Receiver Operating Characteristics (ROC) curve is a graphical representation that illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity), as the discrimination threshold for classifying positive and negative outcomes is varied. A steeper curve and a higher area under the ROC curve (AUC) indicate a better model performance, with the AUC of 1 representing perfect discrimination.

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP | FP |
| Negative | FN | TN |

To calculate true positive rate (sensitivity):

To calculate false positive rate (1-specificity):

We aim for the AUC to be close to 1, and to obtain specificity, sensitivity and accuracy close to 100.

## **Statistical Features**

Statistical features that have been extracted and analyzed throughout this chapter include *mean*, *variance*, *kurtosis*, and *skewness*. Statistical features play an important role in the analysis of the EMG data in this practical session. These features offer valuable insights into the characteristics of the signals.

By analyzing and comparing these statistical features across different EMG signals within our dataset, we can gain a deeper understanding of the variations in muscle contraction levels and distinguish between different classes of muscle activity.

### **Mean**

The *mean* is a fundamental statistical metric. Each *mean* serves to summarize a given group of data, often to better understand the overall value of our EMG dataset. To calculate the *mean* we’ve computed the below formula in MATLAB.

After computing the *mean* for each class, we have displayed it using scatter, differentiating between the classes by using two colors and patterns,

A graph with red lines

Description automatically generated

Figure Mean as Feature Display

Considering that the AUC measures the discriminative power of the feature, in this case *mean* as a feature, we note that 0.27 is relatively low and therefore *mean* doesn’t perform well in distinguishing between the two classes of the dataset. Considering that the AUC of 0.5 would represent random chance, 0.27 means that the feature’s performance is below random chance. We also notice that an accuracy of 0.5 indicates that *mean* as a classifier, is guessing randomly between the two classes instead of effectively differentiating. In this scenario, we can also say that our classifier has an overfitting problem, considering that sensitivity is equal to 1 and the specificity is equal to 0. These values are contradictory with the low accuracy and AUC measurements.

To conclude, the ROC curve and associated metrics, indicate that using the *mean* feature for the classification is not effective in distinguishing between classes.

If we look at the obtained boxplots however, we do get a more discriminative view of our classes. Class 1 has less variability than class 2, this noticed in the much larger size of the box for class 2. This indicates that not only is the central tendency different, but the spread of the data points is also wider. The central tendency difference can also be noticed in the position of the median. The latter also provides us with the information that the two boxplots do not overlap. What’s important to point out is the presence of an outlier in the first class. This outlier can have a significant impact on the AUC and accuracy values on our classification model.

### **Variance**

*Variance* is a statistical measure that quantifies the degree of spread or dispersion in a set of data points. Through it we can tell how each individual data point deviates from the *mean*(average) of the dataset. We used this measure to show how far on average are the data points on a measurement from the mean of that measurement. *Variance* in a set can be calculated by the following formula:

So, for every single one of the measurements, after we have calculated the *mean*, we find the *variance* linked to this *mean*. After calculating this measure for all the given measurements, we have the following picture.

A graph with a red line

Description automatically generated

Figure Variance as Feature Display

As evident from the analysis, *variance* emerges as a highly discriminating feature between the two given classes, achieving an impressive accuracy rate of 95%. Furthermore, the model maintains a remarkable balance, with a sensitivity of 100% and specificity at 90%. Notably, Class 2 in the dataset exhibits a consistently higher *variance* compared to Class 1, indicating greater variability within the former group. This observation is reinforced by the boxplot analysis, which reveals a higher median *variance* for Class 2 in contrast to Class 1. Additionally, the data points within Class 2 exhibit a more significant spread, highlighting the class's higher variability and confirming its distinct characteristics. This higher *variance* on the second class could show a higher degree of contraction than that of the first class, further proving the legitimacy of these calculations.

The discriminative power of variance can also be noticed in the boxplot view. The two boxplots representing each class do not overlap at all between each other, have different whisker lengths and different box sizes. The central tendency and the variability are relatively different from one to the other.

Overall, this feature shows remarkable promise and would definitely be used for creating or enhancing a machine learning model. The amount of the discriminative power it provides together with the knowledge that these *variances* come from different levels of muscle contraction, give this feature great importance.

### **Skewness**

*Skewness* is a measure of the asymmetry of a distribution. *Skewness* measures the asymmetry of the EMG signal's probability distribution. A positive *skewness* suggests that the tail of the distribution extends towards higher values, indicating a prevalence of higher muscle activity levels. On the other side, negative *skewness* indicates a predominance of lower muscle activity levels. To calculate *skewness* we consider the below formula:

Applying the formula in MATLAB we obtain the following figures and measurements.

A graph with red lines and black text

Description automatically generated

Figure Skewness as Feature Display

To analyze *skewness* as a feature and its ability to distinguish between the two classes we look at the ROC curve and the measurements. An AUC of value 0.59 tells us that the discriminative power of *skewness* as a feature is somewhat higher than random chance, but still limited in the ability to distinguish between classes. An accuracy of 0.7 suggests that using *skewness* as a feature allows us to correctly identify and classify 70% of our dataset. The odds are good, but still leaving room for improvement. Seeing how in contradiction we a sensitivity have of 1, where our classifier is excellent at correctly identifying positive cases, but a specificity of 0.5, suggesting that this classifier is not as effective at identifying negative cases. With the misclassification of more than half of the negative cases as positive, we can say that there’s ambiguity and impreciseness for this feature.

The lack of discriminative power for this feature can also be noticed in the boxplot figure. The boxplot of Class 1 fully overlaps with the boxplot of Class 2.

### **Kurtosis**

*Kurtosis* measures the occurrence of outliers in a distribution. In the context of EMG analysis, kurtosis can provide insights into the shape and nature of muscle contractions. High *kurtosis* values suggest sharp, intense contractions, while low *kurtosis* values indicate smoother, more sustained contractions. To calculate we have considered the below formula:

Applying the formula in MATLAB we obtain the following figures and measurements.

A graph with red lines and black text

Description automatically generated

Figure Kurtosis as Feature Display

For this feature we notice that the AUC value is relatively low, only 0.45. An accuracy of 0.65, even though better than random guessing, it still isn’t sufficient for practical applications. Considering the specificity and sensitivity values we also notice that *kurtosis* as a feature is good at identifying the positives, but not very good at correctly identifying the negative classes. The lack of discriminative power for this feature can also be noticed in the boxplot figure. The boxplot of Class 1 fully overlaps with the boxplot of Class 2. We can conclude that *kurtosis* as a feature isn’t a good discriminator for distinguishing between the two classes.

### **Number of Window Variances Within General Variance Range**

We split the entire time signals into 10 identical windows, for each of the windows we calculated the respective variance. The goal is to see how many variances calculated in this way fall within a general interval of confidence. An interval of confidence will be called the calculated overall variance of the signal plus/minus 10%. This was done for all the signals in hope of finding a correlation between the number of variances within range and the class that a signal belongs to. The results were the following:

A graph with a red line

Description automatically generated

Figure Number of Window Variances Within General Variance Range as Feature Display

As can be seen this feature cannot be used to train a machine learning model as it behaves like a random one, with Area Under the Curve being 0.315, accuracy 50% with sensitivity being 100% at the cost of sensibility being 0. There is not much to note about this feature, as the values do not present a pattern in of themselves.

## **3.2 Temporal Features**

This section is going to look through the temporal analysis of the EMG signals given and further testing their reliability just as we did for statistical features on the previous chapter. The temporal features that are of most interest are *energy* and *root mean square*. These measurements are highly dependent on the strength of the signal which in turn is correlated to the amount of contraction a muscle receives.

### **3.2.1 Energy**

Energy is the measure that represents the cumulative magnitude of the signal’s values over a certain amount of time. Through this tool we will be able to test the intensity or strength of the signals retrieved from our dataset over time. To calculate the energy for each of the signals we can use the following formula:

For each of the signals we calculated their energy using the above formula and the results are shown in the picture below.

A graph with a red line

Description automatically generated

Figure Energy as Feature Display

In the graphical representation above, we present a comprehensive analysis of 20 signals, where the first ten signals belong to one class and the second ten to another. The calculated energy values for each signal have yielded compelling insights into the discriminative power of this feature.

The ROC curve showcases the robustness of energy as a feature for classification. With an impressive Area Under the Curve (AUC) of 0.91, energy demonstrates its effectiveness in distinguishing between the two classes. This discriminative power translates into an accuracy rate of 95%, indicating the model's overall performance. Furthermore, the sensitivity of 100% highlights energy's ability to correctly identify all instances of the second class, while the specificity of 90% ensures precise classification of the first class.

Energy calculations reveal a notable trend: the second class consistently exhibits higher energy values compared to the first class. Additionally, the point plots illustrate that the energy values within the second class are more spread out, suggesting a greater variability in signal strengths. While the first class generally exhibits lower energy levels, there is one notable outlier, indicating an exception to this trend.

The boxplot further supports our findings, with a higher median energy value for the second class compared to the first class. Boxplot also confirms the greater variability of the points on the second class showing the wider spread of the data points.

Overall Energy is a great feature to use for a machine learning model as it provides an amazing amount of discriminative power.

### **3.2.2 Root Mean Square**

Root Mean Square (RMS) is a mathematical tool employed to gauge the size or intensity of a signal, especially beneficial when dealing with signals that exhibit temporal variations, such as vibrations. The RMS value serves to articulate the signal's comprehensive "magnitude" or "potency," encompassing both its positive and negative fluctuations.

By using the above equation through all the signals on the dataset we managed to get the following graphical representations.

A graph with a red line

Description automatically generated

Figure RMS as Feature Display

As we can see in the graphical representations of the Root Mean Square, this feature has a lot of discriminative power making it one of the most likely features to be used in a machine learning model. To aid in this conclusion, come the facts that the Area Under the Curve (AUC) of ROC is 0.91, the accuracy is impressive at 95% and sensitivity and sensibility are near perfect with them being 100% and 90% respectively.

These values do not come as a surprise since as we can see the division of data points between classes is highly visible in both the plotting of the points as well as in the boxplot. The median of the second class is bigger than that of the first class. The data points on the second class are also more spread between them than the first class. We have an outlier on the first class, showing that that signal either is faulty and there is some kind of noise. On the other hand, it could show just an exception to the rule.

Overall, the Root Mean Square is a great feature to further develop a machine learning model on, based both on the discriminative power it possesses as well as the logic that the equation is trying to capture.

## **3.3 Spectral Features**

Spectral Features open a whole new world of classification possibilities. The transformation to the frequency domain provides valuable insights and advantages related to signal analysis and in our case, it gives us more space to evaluate our data in. Some features that we are going to see in this domain are *Power*, *Mean Power Frequency*, *Skewness and Kurtosis coefficients*, *Energy of the signal by Frequency Bands* and *High to Low Ratio*. For all the above features we need first to transform our signals and get a representation of it on the frequency domain, more specifically we will need to calculate the Power Spectral Density (PSD) for every signal. To calculate PSD, we used the built-in *pwelch* function on MATLAB.

To get the PSD from an input signal, *pwelch*:

* Segments the input time-domain signal into overlapping or non-overlapping segments (also called windows). Each segment needs to be smaller than the overall signal.
* Applies a window function on each segment to taper the signal at the edges of the segment. The windowing helps reduce spectral leakage, which is an undesirable spreading of energy.
* Applies Fast Fourier Transform (FFT) to each segment.
* Averages the power spectral densities (PSDs) of all the segments.
* Returns the Power Spectral Density of the signal in relation to frequency.

A graph with numbers and lines

Description automatically generated

Figure Power Spectral Density

### **3.3.1 Power**

Power (*P*) in signal processing is mathematically defined as the average rate at which energy is transferred or the rate at which work is done by the signal per unit of time. It is typically measured in units like watts (W) or decibels relative to a reference level (dBm). Power is directly correlated to the energy, intensity of the signals and therefore this feature is interesting and has the potential to be a main feature on a machine learning model. We can calculate the power for a signal using the following formula:

where PSD is the Power Spectral Density of the signal.

After calculating the power for all the signals, we have the following graphical representations.

A graph with red lines and black text

Description automatically generated

Figure Power as Feature Display

After the signals went through power analysis, we can see that the power has a great discriminative power. ROC curve supports this as we have great area under the curve of 0.9, with accuracy 95% and sensibility and specificity respectively being 100% and 90%. These features are justified because there is a direct correlation between energy and the amount of work that energy makes (Power). And it stands to reason that different levels of muscle contractions would generate different energy levels.

As can be seen from the initial plot there is a visible difference between signal instances that belong to Class 1 and Class 2. We can notice that Class 2 on the overall has higher power than Class 1, and this is further backed on the boxplot by the median. We see no visible overlap between the two classes on the boxplot further solidifying the strength of this feature. Class 2 datapoints are also more spread than the datapoints from Class 1, this is further shown on the boxplot as well.

All of this analysis further proves that power is an amazing feature and would bring an amazing amount of discriminative power to any machine learning model.

### **3.3.2 Mean Power Frequency**

Mean Power Frequency represents the frequency at which the signal’s power is concentrated or centered. This measurement provides valuable insights into muscle activity and fatigue. Thanks to it we can assess muscle health. This measure can be calculated thanks to the following formula:

By using the it for all the signals in our dataset we got the following graphical representation for plot, ROC curve and boxplot.

A graph with red lines

Description automatically generated

Figure Mean Power Frequency as Feature Display

From the very first look at the diagram we can see that this feature has a low discriminative power. From the way the points are distributed on the first plot or how the boxplots on the first and second classes overlap, we can say that this feature is just a bit better than a random feature, with Area Under the Curve being 0.62, a good accuracy of 70% and sensitivity of 100% but at a cost of specificity being 40%. From the boxplots again we can tell that the median of both classes almost have the same value and this is further backed up by the almost random distribution of mean power frequencies on the first plot. The data points of the first class seem to be more spread than that of second class, which shows that the energy of the first class of signals is more widespread.

Overall, this feature would not be a good choice for modeling machine learning models with. The almost random AUC together with the extreme loss of specificity would not make it a good discriminator.

### **3.3.2 Moment Coefficient of Skewness**

The skewness of a random variable, is the third standardized moment defined as:

After applying this formula in MATLAB we obtain the following figures and measurements:

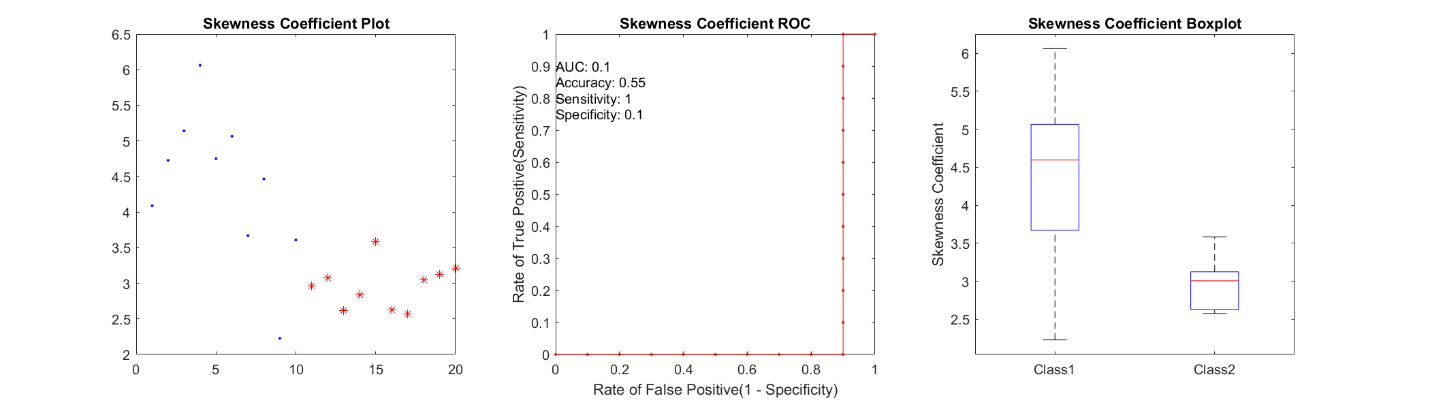


Figure Moment Coefficient of Skewness as Feature Display

From the obtained figures and measurements, we can notice that the overall discriminative power of this feature is relatively weak. Considering the accuracy, we can conclude that this feature, when used for classification, is performing at chance level. In other words, it’s not effectively distinguishing between the two muscle contraction levels. A sensitivity of one for this feature means that we are correctly identifying all instances for the higher muscle contraction levels (EMG11-EMG20). This suggests that the feature is highly sensitive to this class. The results of specificity however show that the feature is not performing well in correctly identifying the lower muscle contraction levels (EMG1-EMG10). This accuracy in distinguishing high muscle contraction levels, but not in distinguishing lower muscle contraction levels, can also be viewed in the boxplot.

To conclude, while the feature appears to be sensitive to the higher muscle contraction level, it is not effective in distinguishing between two classes overall. The low AUC and accuracy, combined with the relatively poor specificity, suggest that this feature may not be suitable for robust classification.

### **3.3.3 Moment Coefficient of Kurtosis**

The coefficient of kurtosis is the average of the fourth power of the standardized deviations from the mean. It is calculated using:

After applying this formula in MATLAB we obtain the following figures and measurements:

A graph with red lines and black text

Description automatically generated with medium confidence

Figure Moment Coefficient of Kurtosis as Feature Display

As we see we have obtained the same results when using moment coefficient of kurtosis as we have done when computing moment coefficient of skewness. This means that both features are performing similarly in the classification task. We can justify this considering that the formula of moment coefficient of skewness expresses skewness in terms of the ratio of the third cumulant to the 1.5th power of the second cumulant and it is analogous to the definition of moment coefficient of kurtosis as the fourth cumulant normalized by the square of the second cumulant. The mathematical expressions used to calculate these features are alike and result in similar discriminative patterns.

### **3.3.4 Energy by Frequency Bands**

Energy by frequency bands refers to the distribution of a signal's energy across different frequency ranges or intervals within the frequency spectrum. It is a way to quantify how much energy is associated with specific frequency components within a signal. We can calculate the energy by frequency bands using the following formula.

The only thing left is to find what are the frequency bands of interest for our data. After some research we managed to find that the most interesting frequency band for EMG is 50 – 150Hz. This range was almost the same value where the most notable peak in PSD could be noticed [Figure 8].

After calculating the energy by frequency bands for the band: 50 – 150 Hz and for all the measurements, we achieved the following results.

A white sheet with red lines and black text

Description automatically generated

As we see this feature is not discriminative between the two classes, the Area Under the Curve for it is 0.6 (almost the same as for a random feature), the accuracy of it is not much better being 65% and with a sensitivity of 90% and specificity of 40%. The data points are all mixed up and it is clear the different levels of muscle contractions do not affect this feature. The median is almost the same between both classes with the only thing differentiating between them is the different spread of the data points.

Overall, this feature is not discriminative enough in order to be used on a machine learning model as its contribution at best can be useless, while at worst detrimental.

### **3.3.4 High/Low Ratio**

This feature is particularly valuable for characterizing the frequency distribution within EMG signals. In this context, the H/L Ratio serves as a powerful tool for quantifying the balance between high-frequency and low-frequency components present in EMG data. We compute this feature using:

Considering the characteristics of our EMG signal, when we plot the Power Spectral Density (PSD) [Figure 8], we can identify the frequency component that contains the most relevant information. For the low-frequency range we have chosen 50 Hz to 500 Hz and for the high-frequency range we have taken 500 – 2000 Hz. The low frequency range is associated with muscle activity and the PSD analysis shows a prominent peak in this range, while the high frequency range captures noise and interference, hence the zero value on our PSD plot. This justifies our chosen frequency bands for high and low.

After computing this in MATLAB we obtain the following figures and measurements:

A white sheet with black text

Description automatically generated with medium confidence

Figure High/Low Ratio as Feature Display

High/Low Ratio measurements indicate that this feature performs well in terms of sensitivity but has an overall performance that is not ideal. The AUC indicates that this feature has a lower than random discriminative power as a classifier and the accuracy is relatively low and misleading. A specificity of 0.3 suggests that the classifier has a relatively low rate of correctly identifying negative cases.

# **Evaluation**