Reconhecimento de Padrões
Inteligência Geoespacial

2022/2023

Project Assignment

Music Genre Classification

1 Background

Music Information Retrieval is a field where sound signals are processed aiming to extract relevant information. One of these relevant pieces of information is genre.

A genre can be described as a class, style or even form that identifies a subset of musical themes. Common genres are:

- blues
- classical
- country
- disco
- hip-hop
- jazz
- metal
- pop
- reggae
- rock

Genre classification can be implemented by following a traditional machine-learning pipeline: music acquisition, audio signal pre-processing, feature extraction, feature reduction/selection and variety.

2 Dataset Description

Consider the dataset available at:

https://www.kaggle.com/datasets/gabrielopecs/gtzan-modified-music-genre-classification.

This dataset was created from 1000 music snippets in one of the abovementioned genres. The *dados.csv* file contains in each line one of the snippets represented by more than 190 features. The last column is the class label that associates a given snippet with a genre. More details on the features and classification results can be found at https://ieeexplore.ieee.org/document/1021072.

In real-world scenarios, nothing is perfect, and identified data problems can be found at https://arxiv.org/abs/1306.1461.

3 Objective

Your task is to develop classifiers for genre discrimination. Consider two scenarios:

- **Scenario A (Binary classification)**: consider a one-vs-all classification;
- **Scenario B (Multiclass classification)**: consider the problem of classifing all genres together.

4 Practical Assignment

4.1 Data Splitting & Standardization

Split your data into two datasets, one for classifier development and the other for testing. For example, select 80% of the data for development and 20% for testing. In the development part, you should perform training and validation, for example, by using k-fold cross-validation, and at the end, select the best classifiers for testing. The testing is future data and should never be used for classifier tunning.

It would be best if you also normalised/standardised your data. Remember that as testing is future data, normalisation/standardisation factors should only be obtained from the development data and applied prospectively in the testing data.

4.2 Feature Selection and Reduction

Some supplied features may be useless, redundant or highly correlated with others. In this phase, you should consider using feature selection and dimensionality reduction techniques and see how they affect the performance of the pattern recognition algorithms. Analyse the distribution of the values of your features and compute the correlation between them. Make sure you know your features! Do not forget to present your findings in the final report.

4.3 Classification

You should be able to design experiences to run the pattern recognition algorithms in the given data and evaluate their results. Define the appropriate performance metrics and justify your choices!

Run the experiments in the development data multiple times to present average results and standard deviations (of the metrics used). In the end, you should be able to choose the best classifier and evaluate them in a testing set .

Do not forget that manually inspecting the predictions of your algorithms can give you precious insights into where they might be failing (and why) and what you can do to improve them (e.g. what makes the algorithm fail in this particular case? what unique characteristic does it have that makes it so hard? how can I make the algorithm better deal with those cases?). Go back and forward to the Feature reduction and Feature Selection phases until you are satisfied with the results. It is a good idea to keep track of the evolution of your algorithm's performance during this process. Try to show these trends in your final report to fundament all the issues involved (choosing parameters, model fit, etc.). For scenario A, you should find which genres are better separated and use domain knowledge to justify your results.

4.4 Development language

You can write your code in your language of choice or use the functions and methods available in Matlab and the Statistical Pattern Recognition STPRTool used in the classes (since you are already familiar with it). The methods used in your work should be described, and the parameters should be discussed. Try out different pattern recognition algorithms. It would be best if you tried to understand how they perform differently in your data.

4.5 Results and Discussion

Present and discuss the final results obtained in your Project assignment. Other authors have already studied this problem. Compare your results with those from other sources—for example, those reported in `https://ieeexplore.ieee.org/document/1021072`.

4.6 Code

You should deliver your software code in MATLAB or any other programming language you used during the project.

Remember to comment on your code. Write also a help section to your code that tells the purpose of the function, usage, and explanation of parameters.

5 Documentation

Write documentation (in Portuguese or English) about your project. The documentation should include a cover page where the authors' course name, project title, date, names and student numbers are mentioned.

Describe the methods used for classification in such detail that the reader could implement the same functions for feature selection/reduction and classification based on your documentation and some essential background in pattern recognition. Always justify your choices, even when they are based on intuition. Do not forget to verify your assumptions! Include classification results with the given data in your documentation. At the end of your documentation, you should list all references used.

5.1 Requirements

The practical assignment is meant to be done in groups of two persons. If someone wants to work alone, this is also possible. Larger groups are not allowed.

5.2 Project Submission & Deadlines

1. **Project First Milestone (Deadline: 21st March 2023!)**
   Deliverables:
   - Data Splitting and normalisation/standardisation;
   - Feature Selection & Reduction (Feature selection and Feature Reduction (PCA & LDA));
   - Minimum Distance classifier, Fisher LDA for Scenario A.

- Code + short report.

2. **Project Final Goal (Deadline: 12th May 2023!)**
   Deliverables:

   - Data Splitting and normalisation/standardisation;
   - Feature Selection & Reduction (Feature selection and Feature Reduction (PCA & LDA));
   - Several classifiers for scenarios A and B;
   - Final Report;
   - Matlab code.

3. **Presentation and Discussion (Last Week classes!)**

Acknowledgments

Credits to Kaggle for data supply and description. Credits to the original author: George Tzanetakis.