Reconhecimento de Padrões
Pattern Recognition

2014/2015

Project Assignment

Higgs boson Identification

1 Background

The Higgs boson is an elementary particle postulated in the 1970 by several scientists, including Peter Higgs (http://en.wikipedia.org/wiki/Higgs_boson). They described that some fundamental particles have mass and stayed together because of a field, created by the so called Higgs boson (Figure 1). Between 2012 and 2013 a particle with some properties of the Higgs boson was measured on behalf of the Atlas project at the CERN Large Hadron Collider (LHC) (http://home.web.cern.ch/topics/large-hadron-collider). The Higgs boson become measurable by colliding two protons that traveled in opposite directions at approximately the speed-of-light.

The observation of the Higgs boson opened new windows in physics, but also new challenges from the computational point-of-view. A huge challenge is to identify collisions where the Higgs boson becomes measurable from a very large amount of experiments.

2 Objective

This assignment is based on the Kaggle Higgs Boson Machine Learning Challenge (https://www.kaggle.com/c/higgs-boson). In this project you will adopt a Pattern Recognition/Machine Learning approach to classify collision events into "tau tau decay of a Higgs boson" versus "background".

Your task is therefore to develop a system that, given a set of features extracted from simulated LHC collision events, will classify them as being related to a Higgs boson or background event. To achieve this goal, several steps of a pattern recognition system should be undertaken such as (i) pre-processing phase (ii) feature reduction (iii) feature selection (iv) pattern recognition techniques (v) experimental test analysis.

The system should be implemented in MATLAB.

3 Practical Assignment

3.1 Dataset Description

The original dataset contains 250000 events, with an ID column, 30 feature columns, a weight column and a label column. The label column discriminates "background" events from events related to the "tau tau decay of a Higgs boson". The weight column was used on behalf of the Kaggle contest to evaluate the results, and will not be used on this project. Details about the features can be found at `https://www.kaggle.com/c/higgs-boson/data`, and at `http://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf`.

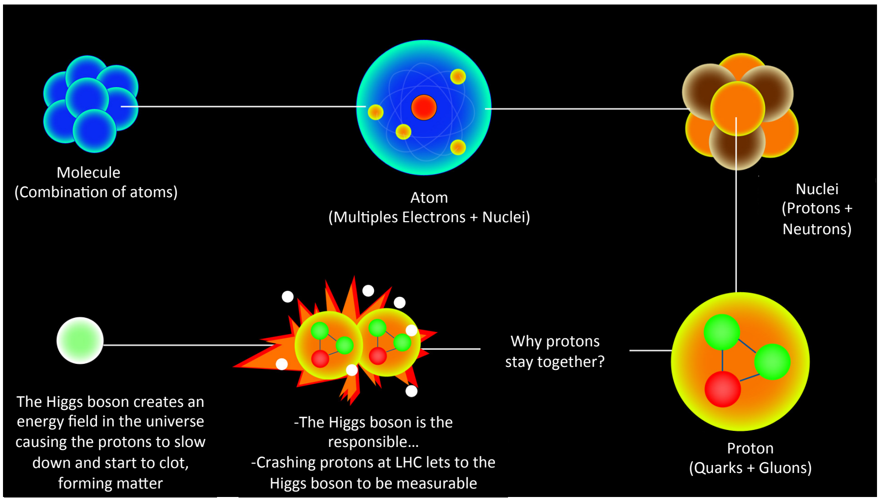

Figure 1: What is the Higgs boson? (Adapted from http://i.imgur.com/arUcX.jpg).

For short, all features are floating point, except PRI_jet_num which is integer. Features prefixed with PRI (for PRImitives) are "raw" quantities about the bunch collision as measured by the LHC detector. Features prefixed with DER (for DERived) are quantities computed from the primitive features, which were selected by the physicists of the ATLAS project. Two main problems characterize the dataset:

- Unbalanced data: ≈66% "background " events versus ≈34% "tau tau decay of a Higgs boson" events.
- Missing data: for some events, some features were meaningless or was not possible to computed them. In this case, the related feature value assumed -999.0, which is outside the normal range of all variables.

For a proper validation, a random selection of 50000 events were removed from the total dataset and will be used to evaluate the performance of your algorithm after the final submission. Thus, just 200000 events will be supplied to develop and test the classifiers. You can download the training and testing dataset from http://eden.dei.uc.pt/~bribeiro/higgs_data.mat.

3.2 Missing Values, Feature Selection and Reduction

Some features present missing values, represent by the value -999.0. Consider to use techniques to estimate plausible values to replace these missing values. Some of the supplied features may be useless, redundant or highly correlated with others. In this phase, you should consider the use feature selection and dimensionality reduction techniques, and see how they affect the performance of the pattern recognition algorithms. Analyze the distribution of the values of your features and compute the correlation between them. Make sure you know your features! Don't forget to present your findings in the final report.

3.3 Experimental Analysis

You should be able to design experiences in order to run the pattern recognition algorithms in the given data and evaluate their results. Keep in mind that this is an unbalanced binary data set. Try to design the classifier taking into account these issues. Justify your assumptions and decisions.

Define the performance metrics to evaluate your method (e.g. (AER) Average Error Rate, F-measure, ROC Curves, etc.). To run the experiments multiple times and to be able to present average results and standard deviations (of the metrics used) you should use cross-validation. Cross-validation is a Matlab function available in STPRTool.

Don't forget that manually inspecting the predictions of your algorithm can give you precious insights of where it is failing and why, and what you can do to improve it (e.g. what makes the algorithm fail in this particular case? what special characteristic does it have that makes it so hard? how can I make the algorithm deal better with those cases?). Go back and forward to the Pre-processing, Feature reduction and Feature Selection phases until you are satisfied with the results. It is a good idea to keep track of evolution of the performance of your algorithm during this process.

3.4 Pattern Recognition Methods

You can write your own code or use the functions and methods available in the Statistical Pattern Recognition STPRTool used in the classes (since you are already familiarized with it). The methods used in your work should be described as well as discussion of the parameters used. Try out different pattern recognition algorithms. You should try to understand how they perform differently in your data.

3.5 Results and Discussion

Present and discuss final results obtained in your Project assignment.

3.6 Code & Graphical User Intreface (GUI)

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

For your project you should write code for a graphical user interface in MATLAB. To aid you, MATLAB has for that purpose the built-in tool "guide" which can be called from the console. The GUI should improve the interaction of the user with the code by providing options for data-loading, feature selection/dimensionality reduction, classification, post-processing, validation and visualization. **Take special attention to the fact that your algorithm will be tested in the validation dataset that is not available for training and testing. Thus, you should implement also the pre-processing steps that lead to the proper algorithm validation.** Remember to comment your code. Write also a help section to your code that tells the purpose of the function, usage, and explanation of parameters. In MATLAB, comments following the first line of a function will show when help command is used with the name of the function.

4 Documentation

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

Describe the methods used for classification in such detail that reader would be able to implement the same kind of functions for feature extraction and classification just based on your documentation and some basic background in pattern recognition. Always justify your choices, even when their are based on intuition. Don't forget to verify your assumptions! Include classification results with the given data to your documentation. At the end of your documentation you should have a list of all references used.

4.1 Requirements

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.

4.2 Project Submission & Deadlines

Project First Milestone: Data Preprocessing (Missing Values, Feature Reduction, Feature Selection), Matlab Code + short report. Deadline: 30th April 2015!
Project Final Goal: Final Report + Matlab GUI. Deadline: 29th January, 2015.
Presentation and Discussion: 1st June 2015.

Acknowledgments

Credits to the Kaggle platfom and to the Atlas project in general.