COMP4124: Big Data – Group Project

Table of Contents

The 3rd UoN Local Conference on Big Data

Nottingham, May 17, 2024 COMP4124: Big Data

– Group Project Brief

Overview

The 5th UoN Local Conference on Big Data calls for novel research projects in the field of Big Data and Machine learning with Apache Spark. The projects will fall into one of these two categories:

- **Technique-based project:** Aiming to enable a machine learning/data mining technique to deal with Big Data. This may include (but it is not limited to):
  - *Data pre-processing*: e.g. feature selection, dimensionality reduction, noise filtering, missing values imputation, etc
  - *Unsupervised learning*: e.g. DBSCAN, hierarchical clustering.
  - *Time series or Regression*: e.g. ARIMA, Logistic regression, SVRs.
  - *Non-standard classification/regression*: e.g. multi-label classification, multi-output, imbalanced classification, semi-supervised learning
- **Problem-based project:** Aiming to exploit some data to solve a problem using a machine learning solution/pipeline for this in Big Data.
  - *Sensor data*: e.g. Temperature of buildings
  - *Social media*: e.g. Twitter messages
  - Project based on a Kaggle competition

The aim of this conference is to create a repository of Big Data solutions from the University of Nottingham. When finished, you are also encouraged to try to publish your code on Spark-Packages!

Below you can find a list of project ideas that are suitable for this. You will have to choose 3 to help with the allocation process, providing your priorities. You could also **suggest your own** project if you like, but you must provide sufficient information about what you will do; in this case, you still have to choose 2 of the projects from the list.

Important dates

- ❑ Expression of Interest: 18t March 2024 (15%)
- ❑ Project allocation: 22nd March 2024
- ❑ First submission deadline: Thursday 28th March 2024 (**Optional**: Non-assessed)
- ❑ Final submission deadline: 13th May 2024 (50%)
- ❑ Presentations: 17th May (15%)

Best paper awards

A certificate will be issued during the Award Ceremony to the group with the best paper in each category.

Organisation

- Armaghan Moemeni, General Chair.
- Weiyao Meng, Session Chair.
- Salim Maaji , Session Chair.
- Ziling Wu, Session Chair.
- Daniel Torres Ruiz, Session Chair.

You are allocated one of the Session Chairs, who will be guiding you throughout the entire process as if they were project supervision meetings. The information about your mentor is in the teams list in Moodle.

Copying Code and Plagiarism

You may freely copy and adapt any of the code samples provided in the lab exercises or lectures. You may freely copy code samples from the Spark documentation, which have many examples explaining how to do specific tasks. This coursework assumes that you will do so and doing so is a part of the coursework. You are therefore not passing someone else's code off as your own, thus doing so does not count as plagiarism. As professional practitioners you must, of course, clearly indicate in your code where you have used or adapted others' code.

You should look at other code/papers on-line, but you need to reference any source/material that you have used as inspiration, and highlight what's your contribution. Turnitin will detect any use of external sources automatically. Successful completion means that you are able to explain your solution during the presentations. The university takes plagiarism extremely seriously and this can result in getting 0 for the group project, the entire module, or potentially much worse.

Getting Help

You MAY ask your module convenor, or any of the lab helpers for help in **understanding the group project requirements** if they are not clear (i.e. *what* you need to achieve).

Talk to us during the labs, after the lecture, or post your questions on Moodle. Any necessary clarifications will then be added to the Moodle page or posted on the discussion forum so that everyone can see them. You may **NOT get help from anybody else (other than your group mates) to actually do the project** (i.e. *how* to do it), including the module convenor or the lab helpers.

We might, however, help each other with technical issues using Apache Spark, but cannot solve your particular parallelisation task. Please use the Discussion Forum to ask any questions about the group project description or issues when using Spark.

Submissions guidelines

1. Expression of Interest (15 marks)

Different project ideas can be found here. These projects are general ideas of what the project could be. However, you should specify what techniques exactly you will focus on to solve the problem. You should read them carefully, and for those topics that you are more interested in, you should read the provided references and get familiar with the topic.

Then, you are asked to provide an *Expression of Interest* for those projects you find more appealing. You have to choose 3, providing priorities, a brief summary of your understanding of the topic, why you are interested, initial thoughts about how to solve them, and identify which particular data mining technique(s) you want to implement in this project and why. You could also suggest your own project if you like, but you should provide sufficient information about what you will do; in this case, you still have to choose 2 projects from the list. Prepare a table like this:

<group-name>.pdf

Priority	Project Code	Thoughts (Max 300 words for each project)
1	OWN	For your own project, you should provide a clear description and references. You should answer: why this is interesting? Why is this Big Data? Where is the data coming from? What are your initial thoughts on how to address it? E.g.: "*Semi-supervised classification methods **[background info]**…. This is interesting because unlabelled data is abundant (typically automatically collected) and easily become very big in many case studies. **[background info]**…. In this project we will focus on multi-view approaches such as Co-Training and Tri-Training (Detailed in paper [1]) because they can parallelise as **[Brief Explanation]**. We plan to investigate different Big Data solutions to parallelise its processing, using multiple base classifiers from Apache Spark. **[Data explanation]** We are going to use standard datasets from the UCI machine learning repository and we are creating artificial semi-supervised datasets taking 2, 5 and 10% of labelled data."* **References:** [1] Paper and/or datasets.
2	PBXX	Provide initial thoughts on how to solve the problem, and any details about the specific algorithms you want to investigate. This doesn't mean you have to stick to those if you later find them not suitable for Big Data.
3	TBXX	Same as before…

If you are proposing your own project, please use the code '**OWN'**. Please do not go for Deep-learning type of projects, as they can be parallelised better on GPUs. Note that we cannot allow many groups to work on the same project, unless specifics have been provided and they are sufficiently different.

Expression of Interest Submission:

- Name your pdf file as *<group-name>*.pdf, replacing *<group-name>* with your group name.
- The file should contain a table with the 3 chosen projects and initial thoughts (up to 300 words for each project).
- The **group leader** must submit the Expression of Interest on Moodle.
- Submission deadline: **Monday 18th March 5pm.**

2. Paper

Papers must be presented in English, **must not** exceed 6 pages, including tables, figures, references and appendixes, in IEEE Computer Society proceedings format as a PDF file.

- **Overleaf –** Click here **(Recommended – use UoN credentials for premium services)**
- **Microsoft Word** (*why do you want to do this?)* – Download here

Expected structure of the paper

Here is the proposed structure for most papers. Whilst you might need to deviate slightly from this structure depending on the project, here are some general expectations.

1- *Title*. A representative name that describes what you have done.
2- *Author names*.
3- *Abstract* (maximum 250 words) to briefly outline in layman's terms the objective or problem statement of the project and includes information on the method, research results, and conclusions of the research.
4- *Introduction and background* (maximum 1 page) present the background to your study, introduces your topic and aims, and gives an overview of the paper, the problem you solved and outline the project contributions. You should aim to include a minimum survey of scholarly sources (such as books, journal articles, and theses) related to your specific topic and/or research question. This aims to link your submission to existing knowledge. **Do not** include a subsection describing Spark, MapReduce or any Big Data technology.
5- *Proposed Methodology* (maximum 1.5 pages) focuses on the reasoning for the certain technique, methods or decision that you made in the context of your study. This describes and explains your solution. It is very important to correlate them to your research questions and/or hypotheses. Please use some diagrams to explain the work/data flow of your solution.
6- *Experimental set-up* (maximum 1 page) indicating the details of the experiments performed. This includes performance metrics, datasets, validation procedure, definition of baselines and any statistical test used.
7- *Results and discussion* (maximum 2 pages) containing a description of the main findings of the research and interpreting the results and discussing the significance of the findings. Include representative diagrams demonstrating (scaling-up) capabilities of the solutions.
8- *Conclusions* (maximum 500 words) allows you to have the final say on the issues you have raised in your paper, to synthesise your thoughts, to demonstrate the importance of your ideas, and to outline a new view of the subject (including a what would you do next?).
9- References: include high-quality references (in-text cited) following the IEEE format. The number should vary depending on the project, around 15-25 references.

A real sample for a problem-based project is provided here, and for a technique-based project here. Note that they are real papers from a few years ago and were obviously not designed for this module but they can give you a flavour of what could be done. I have included some additional annotations about things that might not be needed any more, or other ideas that might be more suitable for your particular project.

***Important**: We are not expecting your project to be publishable material (but maybe it might be!), and these samples are only provided to show examples of document structure. Please read carefully the Detail Assessment Criteria when preparing your manuscript.*

3. Code and Data

The code and some data to test the code must be available on **GitLab**. You don't need to upload an entire big dataset, but smaller datasets should have been used to test the software. All your code should be written in Python 3 and using the Apache Spark library (version > 3.0.1).

Each group has been added to a repository that you should be able to access at GitLab.

The repository should contain **README** file explaining the structure of the repository and how to use it. Any references to other people's code, papers you have used for inspiration must also be indicated here. Some brief documentation should be provided for each code file. You are highly encouraged to actively use the git repository to collaborate as a group.

Optional - First Paper Submission (0 marks)

- Name your pdf file as *<group-name>.*pdf, replacing *<group-name>* with your group name.
- The **group leader** must submit the Paper on Moodle for formative feedback.
- The code won't be included as part of this first submission.
- Submission deadline: **Thursday 28th March 5pm.**

Important note: After submitting the first version of the paper you are expected to continue working on your project addressing any further work you know should be done by the final submission.

4. Final Submission (50 marks)

In addition to the Paper and Code that you are expected to improve from the first submission, the final submission should include an **Individual contribution and peer review statement.** Each member of the group is then expected to complete a survey providing the following information:

- A short personal **contribution and reflection** statement describing the individual contribution to the project and any reflection on how it went (<= 500 words).
- **Peer review assessment**. Assess the other project members by giving them marks out of 10. Be fair and take effort and commitment into account more than actual ability.

Final Paper Submission:

- Name your pdf file as *<group-name>.*pdf
- The **group leader** must submit a single file with the Paper (max 6 pages IEEE format) on Moodle.
- **Each member of the group** must complete the peer review survey and their contribution statement using this link. **You can only submit this survey ONCE**.
- Submission deadline: **Wednesday 13th May 2024 at 5pm.**

6. The conference: oral presentations (15 marks)

All projects will be presented orally at the indicated time in the program (TBA). During your scheduled session time, all **group members will be required to be present** to deliver a

presentation and answer questions from attendees and the Chairs. All oral sessions will follow a similar structure to how they are held in a typical physical conference format.

- The Session Chair will introduce each group.
- The authors will deliver their presentation (**12 minutes maximum**) to the audience.
- Once the presentation has concluded, the Session Chair will facilitate a Q&A period (**5 minutes approx.**) with the audience and the session chair.
- Process repeats for each subsequent paper in the session.

Presentation tips

- Please use the template provided here.
- You should use around 12 slides, 1 slide a minute.
- Start off with a brief introduction of yourselves and the key focus of your paper.
- The outline of the presentation should be similar to the structure of the paper (e.g. introduction and motivation, methodology, experimental set-up, results and conclusions).
- It is a 12-min presentation, do not aim to show every single aspect of your project, focus on the most important things. Key points: Make sure you clearly state your motivation and how novel and good your solution is.
- Be ready for any question and discuss your contribution. You can prepare additional slides for any potential questions you expect. You may be asked to explain your solution, and even show your code, so please make sure that one appointed member of the group is ready to show the paper and the code.
- All members should contribute during the presentation.

Following these tips is important as they are part of the marking criteria.

Program timetable

17th of May – Times to be confirmed. There will be parallel sessions led by each one of the Session Chairs.

Detailed Assessment Criteria for Group Project Assignments

- **Warm-up exercises** *(20 marks): Each group should submit their solutions to the 4 warm-up exercises. Full details provided in the specification documents.*

- **Expression of Interest** (15 marks): *Each group should submit a document containing a table with the 3 chosen projects and initial thoughts (up to 300 words for each project). They can suggest their own project.*
    - **Motivation and initial analysis**
        - Understanding of the chosen topics.
        - Has the team provided enough motivation as to why they want to carry out this research?
        - Are the initial suggestions appropriate for a Big Data project?

- **Final Submission** (50 marks): *Each group should submit a paper-based description of their big data solution together with the code produced.*
    - **Motivation and context**
        - Do the team understand the need for a big data solution for their project?
        - Do they provide a compelling argument as to why is needed?
    - **Design/Methodology**
        - Explanation of the methodology.
        - Is the design efficient? (e.g. does it reduce as much as possible data movement across workers?)
        - Innovation of the proposed methodology. Does it provide a standard divide and conquer strategy, or has the team thought of an elaborated strategy to alleviate the bottleneck of the methods to tackle big data.
    - **Experiments and results**
        - Are the experiments well designed to test the proposed solutions?
        - Do the results support the original motivation?
        - Is the analysis coherent?
    - **Writing**
        - Clear description, reproducibility
        - Quality of visual elements, illustrations, tables.
        - Quality of References
    - **Code – software quality**
        - Efficiency, suitability of Spark operations to solve the problem
        - Documentation
    - **Contribution**: Contribution/reflection statements will feed into the final mark of the project.

- **Group Presentation** (15 marks): *Each group is asked to deliver a 12-min presentation summarising their contribution + 5 mins for Q&A.*
    - Quality and clarity of the presentation
    - Response to questions from the panel and public
    - Understanding of their solution
    - Individual participation in the presentation. All members are expected to participate equally.

Project Ideas

The aim of this group project is to offer you an opportunity to put your hands-on to design/develop a Big Data solution. All the project ideas presented in this section are general descriptions and will allow you to have a preliminary idea of what the project will be, but we are expecting you to do research and determine the final shape of the project.

Important to bear in mind: Even though we have Databricks, we highly encourage you to use subsets of the datasets when testing/designing your solutions. Only run experiments with the full dataset to provide the results for the final submission paper. Also, as a thumb rule, I wouldn't be expecting you to run experiments that take longer than 1 day of execution, unless this is just of a 'sequential' program that you are using as a baseline. Note that creating those subsets might not be trivial in all cases, as you still want them to be representative (i.e. preserve the class distribution!).

Use of Deep Learning: Whilst it is ok to use Deep Learning techniques, this is usually best parallelised using GPUs which is not the goal of this module. You can certainly parallelise Deep Learning on Spark (i.e. using TensorFlow On Spark), but **this is not our goal**.

Best Tips:

- An experimental design is super important when talking about big data. Estimating the potential length of all the experiments you want to carry out is key to control the time. E.g. I would never try to test my code or any minor changes on a 5 million instance dataset.
- Determine the kind of experimental validation need for your topic. E.g. selecting an appropriate performance measure or follow a standard 5-fold cross validation.
- Measure the scalability of your solution (scale-up, size-up, speedup).
- Analyses the effects in performance according to multiple parameters. For example, varying number of maps is very important for the accuracy/error rates in local models, and runtime in all models.

Data: for simplicity, we have downloaded and prepared some datasets that you can use for either problem-based or technique-based projects. **These will be made available on the Workspace in due course.** The data will be read-only and located at /mnt/data/. These are:

- BitcoinHeistData.csv
- poker/
- SUSY.csv
- HIGGS.csv
- ECBDL14/ NB arff format , very large (xz compressed still)
    - Note that due to the sheer size of this data set only the test_set is decompressed for you, it is recommended that you partition and work with this.
    - Further files are available in .arff.xz format if you want to try working with them but this is non-trivial and not expected. Python's lzma library can be used for decompression.
    - As well as the original test_set.arff, test_set.header and test_set.body are available, all are plain text.
- 2020tweets/
- Health-Tweets/
- Climate_twitter.csv

- fraud/
- bookmarks/ NB arff format, multilabel
- ElectricDevices/ NB timeseries
- LSST/ NB arff format (also available as text), timeseries

These are obviously not an exhaustive set of datasets that you can work with but aim to provide resources for you to get started immediately with the Big Data parts of the group project. Most files here are presented as text files in formats such as csv, arff and txt, this is convenient for gathering the data, but for working with the data we recommend looking at scheme preserving file types for holding processed data. This documentation may be of interest.

Problem-based projects

This kind of project focuses on the **application** of Big Data to solve a problem, rather than on developing 'new' machine learning techniques for a general purpose. This doesn't mean that you should only use existing algorithms, but you are expected to solve a problem using a big data solution. This might mean that you need to adapt an existing technique to tackle the problem or you investigate how to use/combine several existing methods to do this.

As you will be dealing with real-world problems, you cannot expect the data to be in the final shape to perform data mining. You will have to model that data in a particular way to be able to do this. This will also include data that is far from perfect, including missing values, noise, irrelevant features/instances or inconsistencies. It is expected that you take that into consideration when designing your solution that could for example be a pipeline of subsequent machine learning techniques to get the final output. Some of the project ideas might allow for various projects, so you need to **clearly indicate what you plan to do in your expression of interest.**

PB01: Predicting Contact Maps in Bioinformatics

Contact map prediction is a bioinformatics problem, and more specifically a protein structure prediction classification task. The ECBLD'14 Big Data competition provided a highly imbalanced dataset to solve this problem. Isaac won this competition using Big Data solutions in Hadoop, which were mostly focused on Data Preprocessing. However, we didn't explore many other classifiers, as at the time we didn't have Apache Spark and its great implementations. Also, we focused on oversampling approaches which are certainly NOT ideal for Big Data (as we generated even more data!). So, the challenge of this project is: can you beat this solution or perform similarly with a simple/better approach? You should indicate in your expression of interest the kind of alternatives you would like to explore.

References:

- http://cruncher.ico2s.org/bdcomp/
- https://github.com/triguero/ROSEFW-RF
- https://www.sciencedirect.com/science/article/abs/pii/S0950705115002130

PB02: A Classification Model for Sentiment analysis of Tweets

Sentiment analysis is a huge topic, and you could focus on any particular application as soon as you can gather enough data. Here we suggest you aim to classify twitter text messages according to hot hashtags for US election 2020, which was proposed recently in Kaggle. If you

are planning to use other data, you should explain in your expression of interest. To tackle this problem, your solution should be able to quickly extract the text data features to highlight the hot keywords which are re-tweeted more than once, and, if working with the suggested problem, apply various machine learning algorithms to understand if there is any correlation between what's going on twitter and the actual results in the elections. Alternatively, you could use other databases, such as the climate sentiment in twitter dataset with suitably adjusted end goals.

References:

- https://www.sciencedirect.com/science/article/pii/S1877050920306669
- https://www.kaggle.com/manchunhui/us-election-2020-tweets
- https://www.kaggle.com/joseguzman/climate-sentiment-in-twitter

PB03: A Classification Model for Healthcare Sentiments on Twitter

Twitter is a commonly used social network for communication and data sharing. Twitter datasets are formed according to various keywords, topics and discussions. This project should aim to propose a classification model to study twitter users' sentiments in healthcare. For this, you need clean the dataset as twitter datasets usually are noisy. Then, you should investigate different classification algorithms for a big tweets dataset, collected from various medias such as BBC, CNN, and so forth. This classification should aim to classify the text messages according to healthcare hot keywords. The result of this classification should be optimised as much as possible.

References:

- https://hcis-journal.springeropen.com/articles/10.1186/s13673-017-0116-3
- https://www.researchgate.net/publication/276456888_Twitter_sentiment_classification_for_measuring_public_health_concerns
- https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter

PB04: A Big Data Classification Solution for Fraud Detection

One of the main topics in banking, finances or insurance companies is Fraud detection. There are many different machine learning algorithms to classify transactions as fraudulent vs non-fraudulent, but some of these systems make some mistakes that can be embarrassing for the customers, if they, for example, get the credit card declined. This is typically modelled as an Imbalanced classification problem. My colleagues from the IEEE Computational Intelligence society organised a Challenge on this in 2019 and provided a real-world dataset available on Kaggle. This project should investigate alternative big data approaches to detect fraud detection. You are welcome to search for additional datasets if you like, but if you intend to you should indicate this in your expression of interest.

References:

- https://www.kaggle.com/c/ieee-fraud-detection

PB05: Data science approaches to improve genetic prediction in Dementia with Lewy bodies.

Contact details : Thomas Goddard - thomas.goddard@nottingham.ac.uk

Dementia with Lewy bodies (DLB) is a prevalent and devastating condition. It is the second most common neurodegenerative dementia, and has a shorter life expectancy, and a greater cost of care that Alzheimer's disease (the most common dementia). No disease-modifying

treatments exist, but several are currently undergoing clinical trials. These will slow disease progression, as no treatments can yet reverse progression. It is therefore important to identify diseases individuals early in pathology, to slow the disease before it is too late. Typical diagnosis relies on symptoms and this is often too late for effective treatment. Pathology can begin 15 years before symptoms show, and the life expectancy after symptomatic diagnosis is about 4 years. Genetics can be used to identify cases of DLB, before symptoms develop, and identify individuals in need of early treatment/monitoring. Genetics are constant (not 100% true) from birth and can be used to predict risk throughout life.

Data: We have access to a dataset of around 4000 individuals (a mixture of DLB cases and healthy yet elderly controls). This has *7 million variants. Variants are the genotypes* of each individual, which can be any of 3 options per variant. When encoded into 0s and 1s, this dataset is 4000 x 21,000,000. Some of these variants are closely related to each other and this will need to be accounted for. We also have age, sex, diagnosis type, and country for these samples.

Objective: Finding a way to split this dataset into training and test, and make an algorithm that can accurately differentiate between DLB and control in the test dataset? This is a classification problem.

Relevant papers:
https://www.nature.com/articles/s41588-021-00785-3
This is the original paper from which the data was generated - it is very heavy genetics - it is more important to appreciate how the data was made and how much there is.

https://www.mdpi.com/2073-4409/13/3/223
This is a review on the area that describes the problem in more detail, see section 4.1. Genetic Prediction for Early Case Identification.

Technique-based Projects

As before, these projects are quite loosely defined. We want you to decide what you want to do within each specific topic. Again, you will find project descriptions that offer more than one project, and you should indicate what you want to do exactly in the expression of interest.

Preprocessing

Big data pre-processing is a 'big' topic. Apart from the topics I define here, you might want to look at this book for other ideas.

TB01: Instance Reduction in Apache Spark

In 2015, Isaac developed a local-based MapReduce solution for Instance Reduction in Big Data (Classification). The aim of these methods is to represent the original training data sets as a reduced (manageable) number of instances. My model was implemented in Apache Hadoop and followed a simple divide-and-conquer approach to allow ANY existing instance reduction technique to be applied in Big Data. In this project, you are asked to implement instance reduction algorithms in Apache Spark. This could consist of replicating what I designed for Apache Hadoop in Spark, so you obtain only local models, or you could aim higher and choose one or more instance selection/generation techniques and design global solutions. Please indicate your choices in the expression of interest. You can follow a similar experimental methodology as in the paper below, although I would recommend starting with smaller datasets.

References:

- Paper
- https://github.com/triguero/MRPR

TB02: Feature selection in Big Data

Feature selection is a very hot topic in machine learning and data mining. It allows to reduce the dimensionality of a problem which is needed to learn appropriate models and reduce the volume of the data. It also allows us to identify relevant features and even establish a ranking of features (very interesting in bioinformatics). There are many approaches to select/rank features: filter, wrapper, and embedded feature selection (e.g. implicit selection performed by a RandomForest). In big data, there are some solutions to deal with this, which are mostly focused on filter approaches and univariate models (because they are faster). In this project, you should investigate different feature selection approaches in the context of Big Data with a high number of features. In your expression of interest, you should indicate the kind of approaches you will focus on and if your target is to reduce the dimensionality or determine a ranking of features.

References:

- Paper
- https://github.com/triguero/MR-EFS
- https://spark-packages.org/package/sramirez/spark-infotheoretic-feature-selection

TB03: Pre-processing techniques for Imbalanced learning in Big Data

One of the funniest things in the context of Big Data is that in many cases we still suffer from data scarcity. This usually happens in real problems like in medicine where we have very skewed datasets. For example, we may easily gather lots of data from heathy patients but to predict rare diseases we might not have sufficient information. Machine learning and data mining technique usually struggle to cope with this imbalanced situation. One way to deal with this is by using pre-processing techniques, which aim to find a better balance of the input data prior to any learning. We have worked a lot on under-sampling and over-sampling

approaches for imbalance datasets in big data, but we usually used traditional methods such as: Random undersampling vs evolutionary undersampling, or Random oversampling vs SMOTE. In this project, you should investigate alternative approaches to pre-process imbalanced classification datasets (e.g. SMOTE-ENN, RUSBoost, Repeated Edited Nearest Neighbours) or you could attempt to deal with multi-class imbalance problems.

References:

- Paper, Paper 2
- https://github.com/triguero/EUS-BigData
- https://github.com/scikit-learn-contrib/imbalanced-learn

Clustering and classification

TB04: Advanced Clustering in Big Data

Clustering techniques are especially important to understanding our data. In the current MLlib API of Apache spark we find classical approaches such as k-means or GMMs. However, we do have many more clustering algorithms such as DBSCAN, BIRCH, spectral clustering, or biclustering which could be very useful in Big Data. Scikit-learn has some good algorithms, but they do not scale-up to Big Data. In this project, you should pick some representative clustering algorithm(s) and design a big data solution for it. You should indicate the algorithms you are interested in the expression of interest. Note that there are some Spark-based implementations of some of them (like DBSCAN), which you should find (as part of your literature review) and determine if they are good or not.

References:

- https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster
- https://github.com/irvingc/dbscan-on-spark

TB05: Time-series classification

In the context of time-series, we are not only interested in predicting the future, but also in classifying them. This is for example particularly useful for speech or activity recognition (based on sensors). Apart from Deep Learning-based approaches, the k-Nearest Neighbours (kNN) algorithm has extensively been used for this, but instead of using the Euclidean distance, it uses something called: Dynamic Time Warping (DTW), which allow us to compute distances between Time Series. In this project, you are asked to investigate the classification of time series in the big data context. Could you enable the classic k-NN with DTW to scale up to big datasets with plenty of very lengthy time-series? If you choose this project, you could focus on the k-NN algorithm, but you could also investigate other alternatives. If so, please indicate this in your expression of interest.

References:

- https://github.com/JMailloH/kNN_IS
- https://github.com/markdregan/K-Nearest-Neighbors-with-Dynamic-Time-Warping
- UCR repository

TB06: (Fuzzy) Rule-based classifiers

For those of you doing COMP4033-FUZZ, this might be of interest. Fuzzy Rule-Based Classification Systems (FRBCSs) are used in multiple applications because their output is usually a set of human-readable rules which might be relatively interpretable, which is a BIG challenge in big data if we may produce too many rules. In the paper below, my colleagues designed a state-of-the-art fuzzy rule-based system for Big Data classification that was focused on a very traditional fuzzy rule-based algorithm: Chi's algorithm. The aim of this project is to investigate the suitability of other (fuzzy or not) rule-based classifiers in the context of Big Data Classification. Note that rules could even be derived from a Decision Tree! In your expression of interest, you should indicate the algorithms you are considering.

References:

- Paper
- https://github.com/melkano/chi-bd