Comparison of several forms of dimension reduction on cuantitative morphological features for normal, abnormal and reactive lymphocyte diferentiation.

Giménez Gredilla, Daniel

December 2017

Contents

1 - Project's ongoing development description

The current state of the project is as follows: development is underway, with a good progress. Dimension reduction techniques have been researched and are being currently applied and fine-tuned. Programming languages and frameworks have been researched and are being put to use in the expected fashion. Output is being taken out of them, and the project has achieved a level of conclussion extraction, critical evaluation of results and refactorisation of protocols after those conclussions.

1.1 - Level of goal accomplishment and planned results

By the end of the second phase of this project, all goals and tasks have been majorly achieved. Some have suffered delays due to technical difficulties. The original milestone planning was as follows:

Table 1: Phase 2 milestones

Deadline	Milestone
28-NOV-2017	Scoring system completely defined
08-DEC-2017	Complete set of workflows applied
18-DEC-2017	Behaviour/comparison conclussions from output elaborated
18-DEC-2017	Monitoring report for Phase 2

All milestones and tasks have been majorly accomplished for the second phase of the project, with the aforementioned technical difficulties and delays.

1.1.2 - Dimension reduction and prediction techniques ongoing results

PCA: The most widely used of all dimension reduction techniques; for the purposes of this project, as of now, it is used as a kind of benchmark against which all others are evaluated.

ICA: Currently giving off accuracy results similars to those shown by **PCA**. Even though accuracy is similar, the processing time and memory requirements of this technique are greater than those of **PCA**, so it still ranks worse for performance.

Factor Analysis: Even though this technique is expected to give off worse classification accuracy than **PCA**, in this project it has shown a remarkable level of output quality. Technical requirements have been shown to be far greater than those of **PCA** and **ICA**, but it presents the researcher with an invaluable output, consisting of synthetic grouping of variables into semantically-bound factors that bely hidden commonalities. Currently being assessed for possible artifacts. In an extended project, it would be interesting to further analyse the relationships between variables revealed by this technique.

Autoencoders: Initially one of the techniques to be evaluated, it has been logistically impossible to do so. The **Stacked Denoising Autoencoders** implementation turned out to require a boolean input; the numeric predictors used in the objective data set are incompatible with this technique, at the risk of either losing an absurd amount of information, or coercing them to boolean values by decomposing them into totally unmanageable amounts of binomial factors.

T-distributed Stochastic Neighbor Embedding: This technique was also revealed to be incompatible with the current project, but in this case, just out of pure technical requirements. **Tsne** is a visualizing, dimension reduction technique that relies heavily on the availability of virtual memory to allocate the generated vectors. For the number of factors handled in this project, it has consumed both the resources offered by a modified *HP Proliant Gen 8* server and an **Amazon Web Services c3.2xlarge** EC2 instance without yielding any result or even being able to finish the process. The measures taken to try and tackle this obstacle are described in more detail in **Section 2.2**, *undertaken unscheduled activities*.

Linear Discriminant Analysis: Given that **SDA** and **Tsna** couldn't be undertaken, this additonal technique has been taken into account. This technique has yielded remarkable results, which will be further analysed in the final report for this project. For the time being, the good results yielded by this technique are cautiously attributed to a good comparimentalisation and grouping of the response variables classes, as this technique puts an emphasis on those classes.

1.1.3 - Programming language application

Programming language - Python: Python has compared to **R** by being more efficient in respect to resources, but less flexible, especially in the area of data structures and compatibility between object classes. It's also more difficult to output as a dynamic report, as the libraries required to do so, specifically **Knitpy**, are just ports of other libraries such as **Knitr** for **R**, still in development and as such, bug-riddled and lacking in parameters. Still the scientific libraries used for this project fulfilled the spirit of this language by being available for use *out of the box*, with many of its functions and methods just ready to output results from a direct input.

Programming language - R: R has proved to be more of a *hands-on-approach* language, having a lot of pre-built available libraries, but also having to sometimes modify, touch or combine packages and functions in order for them to achieve the desired result. This also gives it more power and flexibility, given that the user knows clearly what his goals are. The output of dynamic reports is a whole lot more satisfactory and powerful via **Rmarkdown** and **Knitr**, which supports the use of *LaTeX* mathematical notation and the computation of chunk and inline code on the fly.

1.2 - Change justification (if necessary)

There have been changes to the arranged timetables and planned content, due to partially unforeseen technical shortages. The computing server used for this project showed a significant lack of processing power and RAM when applied to a singular technique (**T-Stochastic Neighbor Embedding**). As such, timetables and tasks have been changed accordingly to allow for an upgrade of the server, including the following:

- Acquisition of hardware (**2 x 8Gb Kignston RAM chips** for a total of 16Gb and a **Xeon multi-thread processor**).
- Installation of such hardware.
- Integration of the new hardware into the protocols.

2 - Relation of undertaken tasks

2.1 - Scheduled activities

2.1.1 - Elaborate a short briefing on the value of prediction accuracy as output by this packages. (4 days, 12 hours equivalent)

State: Complete - In schedule.

2.1.2 - Assess the validity of it for all the packages selected, and, if it is not valid for all of them, extrapolate a valid, normalized scoring system. (6 days, 18 hours equivalent)

State: Complete - In schedule.

2.2.1 - Apply each and every package's or function's workflow to the supplied lymphocyte data. (4 days, 12 hours equivalent)

State: Complete - Out of schedule.

2.3.1 - Present the score output of each of the applications in a user-friendly manner. (3 days, 9 hours equivalent)

State: Complete - Out of schedule.

2.3.2 - Extract behaviour/comparison conclussions from this score output. (4 days, 12 hours equivalent)

State: Complete - Out of schedule.

2.3.3 - Elaborate monitoring report for **Phase 2**. (8 days, 24 hours equivalent)

State: Complete - Out of schedule.

2.2 - Undertaken unscheduled activities

As part of the tasks undertaken, it has been necessary to upgrade the technical resources available to the author. The machine in which the project's computation is run is a **Hewlett Packard Proliant Gen 8** microserver. It has proven reliable and sturdy, but not powerful enough for some of the proceedings. As such, more RAM and a more powerful processor were acquired and the server was modded with them. This process took the best part of two days of work, from November 30th through December 2nd.

3 - Relation of schedule deviations and buffering actions (if applicable) - Chronogram update

In the original planning, possible sources of deviation and obstacles were proposed as:

1. Technical problems: A single, important technical problem has arised when undertaking the second phase of this project: computing power. For some of the dimensional reduction techniques (**PCA** and **ICA**) the processing times and virtual memory requirements were met even under the most stringent parameters. On the other hand, some of the least tried or most resource-heavy techniques (**Stacked Denoising Autoencoders** and **T-Stochastic Neighbor Embedding**) repeatedly hit a technical roof which hindered the progress of this project.

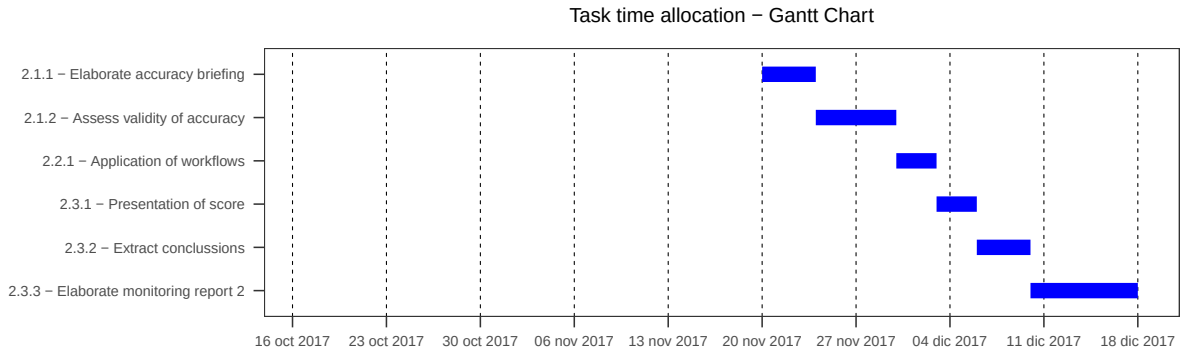
2. Goal overextension: no problems associated with overextension have arised. The scheduled objectives have been deemed by the author as realistic and, apart from other hindrances, achievable.

3. Incompatibilities: no incompatibilities have been found neither in practice nor in literature.

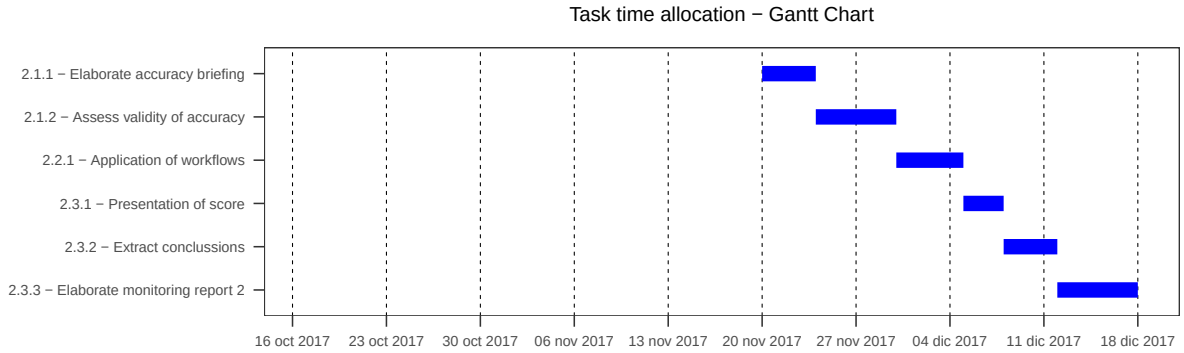
As such, technical buffering measures were undertaken. In the first place, method optimisation techniques were tested, such as the use of resource-light variable structures as turning data frames into sparse matrices or garbage-collection methods after each iteration of the processes, but none of them lowered the requirements in a manner significant enough to be able to complete them to satisfaction.

As this was clearly a hard-cap problem, an **Intel® Xeon® E3-1265L** processor, and two 8Gb **Kingston DIMM 1600 Mhz 240 pins** RAM chips were acquired in order to be able to parallelize processing and provide the machine with a higher virtual memory roof. This microserver modification took the best part of two days of work and brought with it a total blackout in computing during this time. This blackout time was put to use in redaction and problem-solving research.

Referring to the original the Gantt diagram presented within the initial plan:



The planned tasks were slowed down by the aforementioned technical problem. Deviations were mainly focused in **task 2.2.1 - Application of workflows**. The unplanned use of two days was distributed evenly amongst all the following tasks, so the original Gantt diagram is recast as follows:



4 - List of partial results (with attached products)

Comparison suite: Composed of two scripts, one for the **R** implementation and one for the **Python** implementation. Both are coded in dynamic-report frameworks (*R Markdown* for **R** and *Knitpy* for **Python**). Their execution relies heavily on dependencies stated on their respective README documents, so their script outputs are also enclosed in case a potential reviewer is not technically able to run them. Each script produces an intermediate **TeX** file that is then rendered, through a **TeX** engine (*MiKTeX* for Windows, *TeXlive* for Linux), into a final output in PDF format.

Data analysis results: PDF output of the *Comparison Suite*. It is the result of rendering the **TeX** intermediates of both the **R** and **Python** scripts into their respective PDF results.

5 - Relation of goals for the final report

Apart from the redaction of the final report, some additional goals are intended by the author of this project:

- Implementation of an *oversampling-undersampling* hybrid technique for balancing of the dataset.
- Better graphical output: more informative plots for each technique to ensure the reader's visual information is up to the task.

- An additional critical revision and fine-tuning of parameters, to ensure no technical artifact is creating a false output.
- A better dynamic report rendering tool for the **Python** implementation of the comparison script.

6 - Project Tutor's comments