COMPUTATIONAL INTELLIGENCE (CI-MAI)
Guide for the term project, 2016-2017

Lluís A. Belanche, Àngela Nebot
{belanche,angela}@cs.upc.edu

Abstract

This is a brief guide for the correct development of the practical work of the course. As a student, you must study or apply (or both) one or more Computational Intelligence (CI) techniques. These techniques include those specifically covered during the course and any other technique that fits within the CI framework.

1 General information

All students enrolled in CI are required to complete a term project, involving a theoretical work or a practical work to study a real or simulated problem. The project is fairly open-ended and is intended to provide you with a chance to understand fundamental CI techniques or explore recent research on a particular subtopic of CI in more depth. Possible CI techniques include, but are not limited to: swarm intelligence (ACO, PSO, ...), fuzzy computation, neural computation, rough sets, artificial immune systems, belief networks and evolutionary computation.

These are the possible scenarios for your project, in order of preference:

1. Explore a topic that motivates you, as long as it constitutes a valid theoretical, experimental and/or methodological study, focused on a specific CI technique or on a particular application domain solved by one or more CI techniques. In particular, if you have a genuine interest on some problem or you own a data set you wish to explore, then this is a good opportunity.
2. Carry out an in-depth analysis of a specific topic or subtopic, by reading a *small* bunch of *relevant* papers. In this case, you should re-write the material covered in the papers in your own words and eventually reproduce part of the experimental work.
3. Getting a specific kind of project by asking the lecturers. You should specify the kind of work you expect (mainly theoretical, mainly practical or balanced) and the main CI technique involved (e.g., an RBF neural network). This possibility entails a bit of basic research and a certain compromise from the students.
4. Choose one of the prepared (ready-to-go) projects. Unfortunately we have a very limited number of projects to choose from (currently, 6) and they can vary from year to year.

Whatever the case, you are expected to write a complete report describing the work carried out, its motivation, the problems encountered and the solutions envisaged, and the final results and conclusions of the study. The main text is limited to 10 pages[1]. There is a LaTeX template available (and recommended) for use. You can modify it to fit your needs. In case you do not use LaTeX, please try to stick to these general guidelines.

It is expected that you make a proposal for the project for preliminary evaluation. Proposals should be submitted by e-mail (`belanche@cs.upc.edu` and `angela@cs.upc.edu`) as soon as you have them, but no later than **November 2, 2016**. Note this is a mandatory step. Submit your proposal in plain text and be sure to copy all group members on the e-mail so that we can send feedback to everyone. Note this step also applies if you choose one of the prepared projects.

Your project proposal should contain a preliminary project title, a list of team members, and a brief (approximately 2 paragraphs) summary of what you plan to do and why you find it relevant. It should also include a list of the primary references (papers, books) and resources (software, data) that you are going to use.

Important information:

1. Submission of project proposals: **November 2, 2016**.
2. The project will be due **January 23, 2017** (this deadline **will not be extended**)
3. You are expected to form teams of 3/4 people
4. The project is worth 70% of the final grade

2 Project delivery

The final report should include:

1. A description of the work and its goals, and the available data (if applicable)
2. A description of related previous work (if applicable)
3. The CI methods considered, reasoning the choice
4. The results obtained with each method, along with the best set of parameters for each of them (if applicable)
5. Comparison of the results for each method and their significance
6. Scientific and personal conclusions
7. Possible extensions
8. Known strenghts and weaknesses
9. Full code (in separate files, not to be printed!)
10. A brief text file with instructions on how to execute your code (if applicable)

3 Evaluation

The grade will be partly based on the clarity of your report, so please make sure your final report is well organized and clearly written. There should be an introductory part explaining the basics of your work, and a conclusions section, basically stating what you know compared to what was known before the work started; also any gaps, possible extensions or limitations in your development should be noted and explained.

Your work will also be evaluated based on technical quality. This means that the techniques you use should be reasonable, the stated results should be accurate, and technical results should be correct and complete, no matter whether they are your own work or not.

In summary, these are the conditions for a high score (in this order):

[1]This is a strict requirement. If you really need extra space for graphics, tables, etc, place them in a separate appendix.

1. The (good) use of techniques and methods presented in class or those generally accepted to be CI
2. The care and rigor in the (theoretical or experimental) approach for obtaining the results
3. The quality of the obtained results (you should determine and specify which are the appropriate metrics for this)
4. The quality of the written report (conciseness, completeness, clarity).

In addition, as you probably know, there are two generic competences (or *skills*) associated to this course:

1. Effective use of information resources (worth 5% of the final grade)
2. Right attitude towards work (no influence on the final grade, but subjectively evaluated)

This means that you should make an effort to find relevant and high quality information on the topic of your work (books, journal papers, reliable web resources, ...). It also means that all relevant work that you use should be properly acknowledged and/or referenced.

4 Additional information

4.1 CI repositories

Information on research problems, data, code, software:

- Meta-Heuristic Repository:
  `http://www.adaptivebox.net/CILib/misc/prob_link.html`
- Cilib: A library of Computational Intelligence algorithms `http://www.cilib.net/`
- There are also many specific packages (add-ons) for R (`http://cran.r-project.org/`) and Matlab. There is also Knime (`http://www.knime.org/`) (a Python frontend) and the mloss project for machine learning software (you can find almost everything there!) at mloss.org (`http://mloss.org/software/`). Preferred computer languages are Matlab, R, Python and Octave, but any will do.

Additional information on CI methods or on possible problems can be obtained from textbooks, the Wikipedia, other web courses, domain experts, the web ... and yes, from the teachers! (this should be the first option). Please acknowledge properly everything you use.

4.2 Data processing prior to analysis

(this is only applicable to experimental work)

Each problem requires a different approach in what concerns data cleaning and preparation and the selection of the particular information you are going to use can vary; this pre-process is very important because it can have a deep impact on future performance; it can easyly take you a significant part of the time. It is then strongly advised that you analyse well the data before doing anything, in order to gauge the best way to pre-process it. In particular, you shall pay attention to the following aspects (not necessarily in this order):

1. treatment of lost values (missing values)
2. treatment of anomalous values (outliers)
3. treatment of incoherent or incorrect values
4. elimination of irrelevant variables
5. elimination of redundant variables

6. coding of non-continuous or non-ordered variables (nominal or binary)
7. extraction of new variables that can be useful
8. normalization of the variables (e.g. standardization)
9. transformation of the variables (e.g. correction of skewness and/or kurtosis)

4.3 Model selection and honest estimation of performance

(this is only applicable to experimental work)

In accordance with the problem and the available data, you should design a set of experiments based on valid protocols to select models and to estimate the generalization error (or any other measure of future performance) of the final proposed model or solution.

In general, k-fold cross-validation will probably be necessary (the selection of the best value for k is your decision). It is methodologically prohibited to use as test data (i.e., data used for the estimation of generalization error) information that has already been used for the creation, adjustment or selection of the solution.

For a general and in-depth analysis of do's and don'ts concerning experimental work, please consult the course slides ("Tips for experimental work").