Practical assignment 2

Practice comparing computational- and algorithmic-level

Relevant reading
Thagard, P. and Verbeurgt, K. (1998). Coherence as constraint satisfaction. *Cognitive Science, 22*, 1-24.

Software
You are free to code your own implementation in any language you like. We make some Python code available, which you can have a look at and/or build on if you wish. Beware that there is no guarantee that this is completely bug free and any copied mistakes will be your mistakes. Lastly, we make a MatLab implementation available, but beware it was created by someone for purposes unrelated to this course, so it may include functionalities that you will need to ignore. Again, it is not guaranteed to be bug free and you remain responsible for how you use it.

Walk-in hours
Walk-in hours will be held again during regular lecture times on November 15 and 20. Drop by during this time for questions and/or come work together with your group in the lecture room.

Group formation
Make a group of preferably 6 (and minimally 5) students by enrolling yourself (individually) in a group on Brightspace: go to Administration/Groups/View Available groups. Please try to fill the groups in order as much as possible, such that there are no empty groups in between filled ones. Please enroll in a group no later than November 15.

You are free to form the groups yourself, but it is highly recommended to have a group with people with different and complementary backgrounds and skill sets (e.g., if you are a Psychology student, Computer Science student or Erasmus student try to team up with Artificial Intelligence students and vice versa).

You can improve the efficiency of the group by making a balanced division of labor and perhaps appointing one or two coordinators. **Only students who contribute to the group effort can be listed as contributors on the assignment presentation** (taking credit for scientific or academic work that is not your own is fraud and in violation of the guidelines of the code of conduct for research integrity).

If you are not able to find a group or if your group still needs at least one other member, please email the teaching assistant Theo Pijkeren (t.pijkeren@student.ru.nl) with cc to Iris van Rooij (i.vanrooij@psych.ru.nl) to arrange something.

Assignment description
In their paper, Thagard and Verbeurgt (1998) put forth a computational-level theory called Coherence. This computational-level theory is claimed to model many cognitive capacities, including human belief-fixation. In the context of belief-fixation Thagard and Verbeurgt allow for the possibility that special (data) elements in the belief networks have “a degree of acceptability of their own”. Because this aspect of the (informal) theory was not yet captured in the Coherence function, we proposed instead the Discriminating (D-) Coherence and Foundational (F-) Coherence functions as alternative formalizations. Note that Coherence is a restricted version of each of the two functions, giving exactly the same input-output relationship whenever the set of data elements is empty. Because the input/output function defined by Coherence is computationally intractable (and, hence, the same is true for the D/F-Coherence functions), Thagard and Verbeurgt (1998) proposed several inexact algorithms. In this assignment you will systematically investigate the (lack of) fit between the connectionist algorithm (as an algorithmic-level explanation) and the D/F-Coherence function (as a computational-level explanation).

Your assignment includes the following sub-assignments:

- Write or use an implementation of an exact (exhaustive search) algorithm that computes the input/output function defined by D-Coherence (or, if you prefer, FCoherence). Let us call this algorithm ExhCoh.

- Write or use an implementation of the connectionist algorithm, called NCoh, using the default values (see lecture slides).
- Generate a set of instances of the D/F-Coherence problem on which to compare the ExhCoh algorithm and the NCoh algorithm.
- Try to characterize as best as possible the performance of NCoh by comparing its output to that of ExhCoh (e.g., Is NCoh doing better than chance? How does its performance scale?)
- Think about the implications of your findings.
- Think about how the inconsistency between ExhCoh and NCoh could be resolved, if at all. Suggestions can pertain to changes at both the computational- and the algorithmic level theory of coherence. Give motivations for your suggestions.

Some tips and challenges

- The exact algorithm will slow down exponentially for growing inputs. Therefore, it is best to restrict your investigation to relatively small inputs (e.g., < 10).
- You can either generate instances randomly, or deterministically pick a set of instances on which to compare performance of ExhCoh and NCoh. Whichever you do, motivate your choice.
- The primary measure on which to compare ExhCoh and NCoh is the overlap---or better, the lack thereof---in the truth assignments that these algorithms output, and <u>not</u> the coherence value (or "harmony") of the outputs. Think about why.
- Although the coherence value (or "harmony") of the outputs is not the measure of primary interest, you may want to keep track of it for other reasons (e.g., for thinking about the implications of your findings and/or for your suggestions on how to improve the fit between computational- and algorithmic-level explanations of coherence).
- If the default parameter values in the connectionist algorithm do not work for the small instances that you use, you can adjust them slightly (without changing the basic set-up too much). Motivate any changes that you make.
- To prevent activation values in the connectionist values to grow out of bound, you may want to impose strict lower- and upperbounds (-1 and +1).

Presentation instructions

Prepare a maximum 7 minute presentation in which you present your simulation results. Your presentation should cover all of the following (in order of your choice):

- explain any relevant (non-standard) aspects of your implementation of the ExhCoh and NCoh algorithms;
- describe the set of inputs that you selected for the comparison, plus a motivation for choosing this particular set of inputs;
- report your results, giving a clear summary presentation and characterization of outputs of the two algorithms, and the differences observed;
- discuss what you take to be implications of your results;
- present your ideas on how (observed) inconsistencies between ExhCoh and NCoh could be resolved, if at all. Suggestions can pertain to changes at both the computational- and the algorithmic level theory of coherence. Give motivations for your suggestions; and
- discuss anything else you think is relevant (if you have time left!).

Handing-in instructions

Submit your slides (in PDF format) on Brightspace (see Activities/Assignments) **<u>no later than 13:30 on Friday November 22</u>** (i.e., before the presentations start **<u>everyone</u>** should have uploaded their slides, even if you do not yet need to present that day). Only one member of the group has to do this. Submission on Brightspace will lead to your document being stored in TurnItIn. The instructor will retrieve your document from TurnItIn and will post it to Brightspace so that it remains available for viewing for all students for the duration of this course.

Assessment

Your presentations will be assessed on the following dimensions:

- Presentation (style, tempo, clarity of exposition)

- Explanation of implementation
- Set of inputs (explanation of variables, motivation of selection)
- Results (clarity, appropriate and relevant comparisons, interpretation)
- Suggestions for resolving the inconsistency between levels
- Overall insightfulness, soundness and rigor in argumentation and presentation.