University of Glasgow
Artificial Intelligence (H) - COMPSCI 4004 - 2020-2021

v20202021a

The course-work constitutes a total of 30% of the final course grade (∼ 30 hours per student in total). The course-work deals with a single problem but is split between two distinct pieces of work:

- Component A: Group-based exercise worth 25% (∼25h per student) of the final course grade. Marked by the lecturer or a qualified marking assistant. Written feedback will be provided to the group and potentially to the individual student in case individual grades deviate from the group's grade.
  Submission: Friday 4th December 4:30pm via Moodle.
- Component B: Individual video presentation worth 5% (∼ 5h per student) of the final course grade. Marked by the lecturer or a qualified marking assistant. Individual, written feedback based on the marking scheme.
  Submission: Friday 4th December 4:30pm via Moodle.

Please note: The course-work can not be re-done!

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline a) the work will be assessed in the usual way; b) the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands.

---

1 Component A: Group-based assessed exercise

Your group's job is to design, implement, evaluate and document a number of virtual agents which can learn (optimal) COVID-19 mitigation policies.

Environment: The environment is implemented as a OpenAI environment, thus you will need to install and understand the workings of the Open AI Gym environment to be able solve the task. The specific environment/problem under consideration is called ViRL and is an Epidemics Reinforcement Learning Environment for exploring the effect of different mitigation policies on the spread of the COVID-19 virus.

The environment is available here: `https://git.dcs.gla.ac.uk/SebastianStein/virl`. There is a demo available in the notebook folder which shows you how to instantiate and use the environment. Any code update/bug fixes will be committed to this Git repo. We recommend you (as a group) explore and analyse the environment by inspecting the code and environment through the use of deterministic and random policies/agents to understand the properties of the state-space, actions and rewards.

You must design agents for for several instances of the environment mainly focusing on the basic setting where you instantiate the environment for example as follows

```
env = virl.Epidemic(problem_id = 0, noisy=False).
```

You must evaluate your agents for `problem_id=[0:9]`, and with/without noisy observations `noisy={False, True}`. Finally, for full marks, you must also consider/evaluate (but not necessarily design) your agents in the noisy and fully stochastic setting, i.e. where you instantiate the environment with `(stochastic=True, noisy=True)` to evaluate the performance in noisy and changing environments.

Group size: We are flexible regarding group size although we highly recommend you carry out the work in a group of 3-4 students. Groups deviating from this recommendation must be pre-approved by the course coordinator. You can only in ***very exceptional circumstances*** be allowed to do the assignment on your own. Regardless of the group size, you should think about how the tasks associated with the development of your team's submission should be divided. Try to ensure that activities are assigned so that every member of the team can be involved at all times.

Tasks: Your group must design, implement, evaluate and compare a number of agent types. The requirements for the group-work depend on the group size and everyone in the group will be evaluated carefully based on their specific contribution.

Your group must document the problem, your findings and agents in a technical report accompanied by the actual implementation/code and evaluation scripts in a (private) GitHub repository.

	Group Size	**Expected number of people and hours**
Agents:		
Random agent	any	all; problem comprehension; 1h per student
Deterministic agents/policies	any	all; problem comprehension; 2h per student
Q-learning with neural network function approximation	any	15-18h for one student
Policy search with tabular methods using discretized state-space	>1	15-18h for one student
Q-learning with tabular methods using discretized state-space	>2	15-18h for one student
Policy search with linear function approximation (e.g. linear functions, RBF)	>3	15-18h for one student
Policy search with non-linear function approximation (e.g. neural networks)	>4	15-18h for one student
Method of your choice (e.g. evolution / stochastic search-based agents; ask for advice)	6	15-18h for one student
Reporting	any	all; 5-7h per student

Table 1: Specification of agents.

1.1 Details of the submission

1.1.1 Implementation & Code

Your implementation - containing the core implementation of the different agents (along with any dependencies, except the Open AI Gym) - should be uploaded to Moodle as a zip-file containing the source code along with a link/invite to the private GitHub repository.

You must provide the following **executable** files (from the command line or as a notebook):

- `run_random.py/ipynb` an implementation and evaluation for the random agent.
- `run_deterministic.py/ipynb`, an implementation and evaluation for the deterministic agent/policy.
- `run_[agent_name].py/ipynb` files (e.g. `run_qlearnnn.py`) implementing and evaluating a specific RL agent.
- Any custom helper/utility functions you use (do not include standard Python packages).

Each of the agent run-scripts should include:

- Training phase: any required training for the agent (including possible repetitions/episodes of the problem).
- Evaluation phase: an implementation of your evaluation strategy for the (trained) agent. The output can be any results you use in your report or - more likely - results used in a comparison against other agents (i.e. input to the `run_eval.py/ipynb` function specified below). It is generally advisable to save/log any information you may need later for visualisation purposes (as images, txt, csv or binary).
  Note: you can of course call external code/functions from the `run_[agent_name]` file such as common evaluation strategies, however the run-file should serve as the main entry point for the specific agent and be easy to understand and execute.

The above requirements are in place to make sure you separate the individual agents; however, there is still significant freedom in defining the structure and interfaces to your Python modules; so think carefully about and a agree on a common design pattern in your group. Further guidance may be provided on Moodle if deemed necessary.

1.1.2 Experiment & Evaluation

An important aspect of AI is assessing and comparing the performance of agents and different policies. To document the behavior of your agents you should design and execute a suitable set of (computer) experiments which produces a relevant set of graphs/tables to document the behavior of your agents.

Relevant metrics include (but are not limited to): average performance (e.g. reward) vs number of episodes, average performance (e.g. reward) vs steps/actions, convergence/learning rate, etc. There are many possibilities and we encourage you to be rigorous and creative in visualising and documenting your results.

The overall evaluation should be implemented in a single notebook / Python script `run_eval.py/ipynb` which runs your entire evaluation, that is, it should call your agents, collate the results and produce (i.e. save on disk!) the figures/tables you have included in your report. You may cache intermediate results from your individual training phases but the policy evaluation part should be executable directly from the "`run_eval`" script/notebook.

1.1.3 Report

You should document your work and results in a technical paper (max $6+2\times N_{group\,members}$ pages, double-column using the assigned template with suggestion for a structure); including the primary figures, tables, captions but excluding the references and appendices.

The result section of your report must at least include (generated by the `run_eval` script):

- figure/table summarising the results over the specified instances of the problem (you should decide on the metrics)
- figure(s)/table(s) reporting on the learning behavior and convergence of the agents.
- figure(s)/table(s) which allows a rigours comparison of the various agents in terms of performance **after** training.
  Note: All of these aspects can possibly be included in a single figure/table .We encourage you to be creative in generating suitable visualisations etc.

Appendices may be used to provide extra information to support the data and arguments in the main document, e.g., detailed simulation results but should not provide crucial information required to understand the principle of the methods and the experimental results. You can include as many references as you see fit. The report should be submitted via Moodle as a pdf file alongside your implementation and evaluation scripts.

1.2 Marking Scheme - group component

The assessment of the group-based-component is based on the degree (0-100) to which your group's submission (implementation, evaluation scripts and report) ***concisely***, ***correctly*** and ***completely*** addresses the following aspects (percentages are the relative contributions towards the total weight of this components, i.e. 25% of the final grade) :

Pct.	Marking component
10%	**Analysis:** Introduction/motivation and PEAS analysis (including a formal task environment characterisation).
10%	**Method/design:** Presentation of relevant theory and methods (for all your agents) with proper use of mathematical equations, citations and justification for investigating specific methods (with reference to your PEAS/task environment analysis).
20%	**Implementation of the agents:** - The code for all the agents (specified in table 1 must be committed to a private Github repo. We assume at least a few members in your group have worked with GitHub before - if that is not the case then please ask for advice (you can use your student email to get free access to a private repo). - The code must be well-documented, well-structured and follow any naming convention outline above. - The report must contain a presentation of all relevant aspects of the implementation or clearly reference external sources explaining said aspects.
45%	**Experiments / Evaluation:** - Presentation and justification of a suitable evaluation strategy and metrics. - Notebook or code to reproduce the experiment results (i.e. graphs/tables) adhering to the specified requirements. - A suitable presentation and comparison of the performance of the agents (e.g. using graphs/tables) across the various instances of the problem.
5%	**Discussion, reflections, conclusion and suggestions for future work**

- The weighting of the random and deterministic is 5% in total for all marking components in the table above; the remaining 95% is distribution uniformly across the learning agents.
- Submissions not accounting for or not reporting on the `noise=True` condition will be deducted up to 10% per criterion (proportional to the number of affected agents).
- Submissions not evaluating the agents in the `stochastic=True` condition will be deducted up to 5% per marking component (proportional to the number of affected agents).

Your group will initially be awarded a grade based on the submission after which the following aspects will be considered when assessing your individual performance for this component:

- Deltas: In any group it is recognised that people will contribute in different ways. It is important to ensure that you are always aware of your role and that you have an opportunity to make a meaningful contribution to the project at all times. However, for some projects, it is the case that some people contribute more than others, and with this in mind we will be using deltas as a way of adjusting the group mark in order to arrive at an individual's mark for the group-based components of the course.

  A delta typically adjusts the team mark up or down by 0 or more bands for a given individual according to the contribution. The computation of these deltas will be informed by the percentage scores that each member of the team will provide, which gives a numerical estimate of the proportion of the overall effort undertaken by each person (including themselves).
- GitHub activity (i.e. have you provided code, bug fixes, documentation, code reviews or similar). If the lecturer has any doubt about an individual's contribution, they may refer to the project's Github commit logs. This is reason to ensure that you commit often!
- Interview: If there is any remaining doubt and/or conflict, interviews will be conducted with each team member.

1.3 Collaboration and Plagiarism

Discussions and collaboration related to all aspects of the problem is obviously strongly encouraged within the group. Automatic plagiarism checks will be carried out across all the submissions to ensure that there is no plagiarism amongst the groups. Plagiarism among groups or from third parties will be reported to the School and potentially the University Senate.

2 Component B: Individual Presentation (5% of final grade)

Your task is to make an individual video presentation with a duration of approximately 5 minutes (minimum 3 minutes, and maximum 6 minutes) with a suitable choice of visual aids.

You must first decide on a suitable method for learning a policy for the ViRL problem and demonstrate how the agents works. The method can be any of the method you have worked on in the group-work or an entirely different one. Specifically, the video presentation must address the following aspects:

- Description and analysis of the core problem and environment.
- Description of the chosen method and clear justification for the method you have chosen.
- Demonstration of your chosen method/solution and comparison against a baseline (at least one) of your choice e.g. the random agent.
- Reflection on the use of AI-based agents and simulation environments to determine COVID mitigation strategies.

Note: this task is much easier if you have done the group component of the course-work but it can in principle be done independently of the group-work.

2.1 Submission

Your presentation must be submitted via Moodle as a single video file (standard video format). You can use for example Zoom or OBS Studio to record the video. Feel free to using editing tools to improve the recording (there are various free and commercial video editing tools available).

2.2 Marking Scheme: Individual presentation

Your individual presentation is marked separately from the group work and is based on to which degree your presentation ***concisely***, ***correctly*** and ***completely*** addresses the above mentioned aspects according the detailed marking scheme presented overleaf.

3 Collaboration and Plagiarism

For the individual presentation you must submit a fully independent piece of work (i.e. the presentation). Plagiarism checks will be conducted across all submitted videos. Plagiarism will be reported to the School of Computing Science and potentially the University Senate.

Band	**Description**
A1-A5 Excellent	Excellent structure and balance among the required aspects. Very attractive and informative slides, communicating effectively a summary of the key points to the viewers. Fluent, confident delivery. Flowing narrative from one topic to next. An excellent, complete and correct description of all the aspects covering all important points. For A1 and A2: presents aspect of the method or solution not discussed in class and demonstrates an exceptional understanding and insight.
B1-B3 Very good	Very good structure and balance among the required aspects. Slides are informative and attractive, and mostly succeed in communicating effectively a summary of the key points to the viewers. Mostly fluent, confident delivery. Adequate narrative flow from one topic to next. Very good, mostly complete and correct description of all the aspects.
C1-C3 Good	Good structure and balance among the required aspects. Slides are informative and generally succeed in communicating effectively a summary of the key points to the viewers. Hesitant delivery. Disjointed narrative flow from one topic to next. Good description of the various aspects.
D1-D3 Satisfactory	Adequately structured but too much focus on one or a few of the aspects. Slides are satisfactory, though some of the key issues are lost to the viewer, perhaps because there are too many points to be covered in the time available, or because the slides do not contain enough information. Adequate delivery. Little narrative flow from one topic to next. Adequate description of the various aspects. with some limitations/misunderstandings or lack of detail.
E1-E3 Weak	The presentation focuses far too much focus on one or a few of the aspects. Slides are weak, with the result that the viewer is confused as to the main points being made. Halting delivery. Very little narrative flow from one topic to next. Little description/coverage of the various aspects and/or with some misunderstandings or lack of detail.
F1-F3 Poor G1-G2 Very Poor	The content of the talk is minimal or largely inappropriate. Slides are very poor or non-existent, giving very little benefit to the viewers. Incoherent, disorganized delivery. No narrative flow from one topic to next. Very little description of technical contribution. Only very simple description of the problem, method or solution and/or with significant misunderstandings or lack of detail.
H	No positive qualities (or no submission)