Board of Studies

Course Proposal Template

PROPOSED COURSE TITLE: Algorithmic Foundations of Data Science

PROPOSER(S): He Sun

DATE: 23 February 2018

SUMMARY

This template contains the following sections, which should be prepared roughly in the order in which they appear (to avoid spending too much time on preparation of proposals that are unlikely to be approved):

1. Case for Support
– To be supplied by the proposer and shown to the BoS Academic Secretary prior to preparation of an in-depth course description

1a. Overall contribution to teaching portfolio

1b. Target audience and expected demand

1c. Relation to existing curriculum

1d. Resources

2. Course descriptor
- This is the official course documentation that will be published if the course is approved, ITO and the BoS Academic Secretary can assist in its preparation

3. Course materials
- These should be prepared once the Board meeting at which the proposal will be discussed has been specified

3a. Sample exam question

3b. Sample coursework specification

3c. Sample tutorial/lab sheet question

3d. Any other relevant materials

4. Course management

- This information can be compiled in parallel to the elicitation of comments for section 5.

4a. Course information and publicity

4b. Feedback

4c. Management of teaching delivery

5. Comments

- To be collected by the proposer in good time before the actual BoS meeting and included as received

5a. Year Organiser Comments

5b. Degree Programme Co-Ordinators

5c. BoS Academic Secretary

[Guidance in square brackets below each item. Please also refer to the guidance for new course proposals at http://www.inf.ed.ac.uk/student-services/committees/board-of-studies/course-proposal-guidelines. Examples of previous course proposal submissions are available on the past meetings page http://web.inf.ed.ac.uk/infweb/admin/committees/bos/meetings-directory.]

SECTION 1 – CASE FOR SUPPORT

[This section should summarise why the new course is needed, how it fits with the existing course portfolio, the curricula of our Degree Programmes, and delivery of teaching for the different years it would affect.]

1a. Overall contribution to teaching portfolio

[Explain what motivates the course proposal, e.g. an emergent or maturing research area, a previous course having become outdated or inappropriate in other ways, novel research activity or newly acquired expertise in the School, offerings of our competitors.]

Data Science and Big Data Analysis is the research field that studies efficient processing and analysing massive datasets, and has received a lot of attention from different research communities, including algorithms, machine learning, network science, probability theory and statistics. This line of research also motivates a sequence of novel techniques for designing efficient algorithms for massive graphs, and has enormous industrial impact.

However, despite its importance, our school does not have a course discussing the techniques which have been developed in recent years and formed the algorithmic foundations of data science. To fill this gap, the proposed course will provide students with an introduction to these algorithmic techniques, and their applications in data science.

1b. Target audience and expected demand

[Describe the type of student the course would appeal to in terms of background, level of ability, and interests, and the expected class size for the course based on anticipated demand. A good justification would include some evidence, e.g. by referring to projects in an area, class sizes in similar courses, employer demand for the skills taught in the course, etc.]

The target audiences of the course are the students interested in algorithms, machine learning, and data science. More specifically, the course is suitable to students in the programs of Artificial Intelligence (BSc, MSc), Computer Science (BSc, MSc), Informatics (MInf, MSc), and Data Science (MSc).

The expected class size of the course is 150 students, based on the fact that (1) currently the largest theory-oriented course in level 11 (Algorithmic Game Theory and Applications) has about 100 students; (2) a similar but more applied-oriented course (Extreme Computing) has around 150 students including visiting students and students from other schools; (3) the course requires a significant amount of mathematical reasoning, and (4) this is an entirely new course.

1c. Relation to existing curriculum

[This section should describe how the proposed course relates to existing courses, programmes, years of study, and specialisms. Every new course should make an important contribution to the delivery of our Degree Programmes, which are described at http://www.drps.ed.ac.uk/15-16/dpt/drps_inf.htm.

Please name the Programmes the course will contribute to, and justify its contribution in relation to courses already available within those programmes. For courses available to MSc students, describe which specialism(s) the course should be listed under (see http://web.inf.ed.ac.uk/infweb/student-services/ito/students/taught-msc-2015/programme-guide/specialist-areas), and what its significance for the specialism would be. Comment on the fit of the proposed course with the structure of academic years for which it should be offered. This is described in the Year Guides linked from http://web.inf.ed.ac.uk/infweb/student-services/ito/students.]

The course discusses fundamental algorithmic techniques for processing and analysing massive datasets, and closely relates to several existing courses, including Randomness and Computation (INFR11089), Machine Learning and Pattern Recognition (INFR11130). However, despite some overlap with these courses (hashing, clustering), the main focus of the proposed course is to apply basic algebraic (eigenvalues/eigenvectors) and probabilistic (sketching, sampling) tools to design efficient algorithms. From this aspect, this course will provide students with a bridge between mathematical modelling and solving real-world problems in data science.

1d. Resources

[While course approvals do not anticipate the School's decision that a course will actually be taught in any given year, it is important to describe what resources would be required if it were run. Please describe how much lecturing, tutoring, exam preparation and marking effort will be required in steady state, and any additional resources that will be required to set the course up for the first time. Please make sure that you provide estimates relative to class size if there are natural limits to its scalability (e.g. due to equipment or space requirements). Describe the profile of the course team, including lecturer, tutors, markers, and their required background. Where possible, identify a set of specific lecturers who have confirmed that they would either like to teach this course apart from the proposer, or who could teach the course in principle. It is useful to include ideas and suggestions for potential teaching duty re-allocation (e.g. through course sharing, discontinuation of an existing course, voluntary teaching over and above normal teaching duties) to be taken into account when resourcing decisions are made.]

- We expect that about 150 students will register for the course, so a lecture hall with reasonable size is needed.
- The planed course team consists of Dr. He Sun (lecturer), and his PhD student as the TA. Both of the proposed lecturer and the TA have research experience in algorithms for massive datasets.

SECTION 2 – COURSE DESCRIPTOR

[This is the official course descriptor that will be published by the University and serves as the authoritative source of information about the course for student via DRPS and PATH. Current course descriptions in the EUCLID Course Catalogue are available at www.euclid.ed.ac.uk under 'DPTs and Courses', searching for courses beginning 'INFR'*]*

2a. Course Title *[Name of the course.]*:

Algorithmic Foundations of Data Science

2b. SCQF Credit Points:

[The Scottish Credit and Qualifications Framework specifies where each training component provided by educational institutions fits into the national education system. Credit points per course are normally 10 or 20, and a student normally enrols for 60 credits per semester. For those familiar with the ECTS system, one ECTS credit is equivalent to 2 SCQF credits. See also http://www.scqf.org.uk/The%20Framework/Credit%20Points.]

10

SCQF Credit Level:

[These levels correspond to different levels of skills and outcomes, see http://www.sqa.org.uk/files_ccc/SCQF-LevelDescriptors.pdf At University level, Year 1/2 courses are normally level 8, Year 3 can be level 9 or 10, Year 4 10 or 11, and Year 5/MSc have to be level 11. MSc programmes may permit a small number (up to 30 credits overall) of level 9 or 10 courses.]

11

Normal Year Taken: 1/2/3/4/5/MSc

[While a course may be available for more than one year, this should specify when it is normally taken by a student. "5" here indicates the fifth year of undergraduate Masters programmes such as MInf.]

4

Also available in years: 1/2/3/4/5/MSc

Different options are possible depending on the choice of SCQF Credit Level above: for level 9, you should specify if the course is for 3rd year undergraduates only, or also open to MSc students (default); for level 10, you should specify if the course is available to 3rd year and 4th year undergraduates (default), 4th year undergraduates only, and whether it should be open to MSc students; for level 11, a course can be available to 4th and 5th year undergraduates and MSc students (default), to 5th year undergraduates and MSc students, or to MSc students only]

5, Msc

2c. Subject Area and Specialism Classification:

[Any combination of Computer Science, Artificial Intelligence, Software Engineering and/or Cognitive Science as appropriate. For courses available to MSc students, please also specify the relevant MSc specialist area (to be found in the online MSc Year Guide at http://web.inf.ed.ac.uk/infweb/student-services/ito/students/taught-msc-2015/programme-guide/specialist-areas), distinguishing between whether the course should be considered as "core" or "optional" for the respective specialist area.]

Computer Science, Artificial Intelligence

Relevant MSc specialist area: Theoretical Computer Science (core), Machine Learning (optional)

Appropriate/Important for the Following Degree Programmes:

[Please check against programmes from http://www.drps.ed.ac.uk/15-16/dpt/drps_inf.htm to determine any specific programmes for which the course would be relevant (in many cases, information about the Subject Area classification above will be sufficient and specific programmes do not have to be specified). Some courses may be specifically designed for non-Informatics students or with students with a specific profile as a potential audience, please describe this here if appropriate.]

See Subject Area above.

Timetabling Information:

[Provide details on the semester the course should be offered in, specifying any timetabling constraints to be considered (e.g. overlap of popular combinations, other specialism courses, external courses etc).]

Considering the balance of the courses at a similar level offered in the two semesters, this course should be offered in Semester 1.

No timetabling constraints have been found.

2d. Summary Course Description:

[Provide a brief official description of the course, around 100 words. This should be worded in a student-friendly way, it is the part of the descriptor a student is most likely to read.]

The course aims to introduce algorithmic techniques that form the foundations of processing and analysing massive data sets of various forms. In particular, the course discusses how to pre-process massive data sets, efficiently store massive data sets, design fast algorithms for massive data sets, and analyse the performance of the designed algorithm. Through various examples and the coursework, the students will see applications of the topics discussed in class in other areas of computer science, e.g., machine learning, and network science.

Course Description:

[Provide an academic description, an outline of the content covered by the course and a description of the learning experience students can expect to get. See guidance notes at: http://www.studentsystems.is.ed.ac.uk/staff/Support/User_Guides/CCAM/CCAM_Information_Captured.html

The course is to discuss algorithmic techniques that form the foundations of processing and analysing massive data sets of various forms. Specific techniques covered in the course include effective representation of data sets, extracting useful information from a dataset based on algebraic tools, designing faster algorithms based on sampling and sketching techniques. Students in class will learn these techniques through intuitions, theoretical reasoning, and practical examples.

Pre-Requisite Courses:

[Specify any courses that a student must have taken to be permitted to take this course. Pre-requisites listed in this section can only be waived by special permission from the School's Curriculum Approval Officer, so they should be treated as "must-have". By default, you may assume that any student who will register for the course has taken those courses compulsory for the degree for which the course is listed in previous years. Please include the FULL course name and course code].

It is RECOMMENDED that students have passed Algorithms and Data Structures (INFR10052).

Co-Requisite Courses:

[Specify any courses that should be taken in parallel with the existing course. Note that this leads to a timetabling constraint that should be mentioned elsewhere in the proposal. Please include the FULL course name and course code].

None

Prohibited Combinations:

[Specify any courses that should not be taken in combination with the proposed course. Please include the FULL course name and course code].

None

Other Requirements:

[Please list any further background students should have, including, for example, mathematical skills, programming ability, experimentation/lab experience, etc. It is important to consider that unless there are formal prerequisites for participation in a course, other Schools can register their students onto our courses, so it is important to be clear in this section. If you want to only permit this by special permission, a statement like "Successful completion of Year X of an Informatics Single or Combined Honours Degree, or equivalent by permission of the School." can be included.]

This course has the following mathematics prerequisites:

1 Calculus: limits, sums, integration, differentiation, recurrence relations

2 Graph theory: graphs, digraphs, trees

3 Probability: random variables, expectation, variance, Markov's inequality, Chebychev's inequality

4 Linear algebra: vectors, matrices, eigenvectors and eigenvalues, rank

5 Students should be familiar with the definition and use of big-O notation, and must be comfortable both reading and constructing mathematical proofs using various methods such as proof by induction and proof by contradiction.

Available to Visiting Students: Yes/No

[Provide a justification if the answer is No.]

Yes.

2e. Summary of Intended Learning Outcomes (MAXIMUM OF 5):

[List the learning outcomes of the course, emphasising what the impact of the course will be on an individual who successfully completes it, rather than the activity that will lead to this outcome. Further guidance is available from https://canvas.instructure.com/courses/801386/files/24062695]

On completion of this course, the student will be able to

1. Demonstrate familiarity with fundamentals for processing massive datasets.

2. Describe and compare the various algorithmic design techniques covered in the syllabus to process massive datasets.

3. Apply the learned techniques to design efficient algorithms for massive datasets.

4. Apply basic knowledge in linear algebra and probability theory to prove the efficiency of the designed algorithm.

5. Use appropriate software to solve certain algorithmic problems for a given data set.

Assessment Information

[Provide a description of all types of assessment that will be used in the course (e.g. written exam, oral presentation, essay, programming practical, etc) and how each of them will assess the intended learning outcomes listed above. Where coursework involves group work, it is important to remember that every student has to be assessed individually for their contribution to any jointly produced piece of work. Please include any minimum requirements for assessment components e.g. student must pass all individual pieces of assessment as well as course overall].

The course assessment consists of a written exam, and a course work.

The written exam is to test a student's understanding about the algorithms' design and analysis techniques discussed in class, as well as a student's ability to apply the learned techniques to design and analyse new algorithms. This corresponds to the Intended Learning Outcomes 1-4.

The coursework is to test a student's ability to solve more complicated algorithmic problems occurring in practice, and use appropriate software to analyse massive data sets. This corresponds to the Intended Learning Outcomes 3-5.

Assessment Weightings:

Written Examination: _75_%

Practical Examination: _0_%

Coursework: _25_%

Time spend on assignments:

[Weightings up to a 70/30 split between exam and coursework are considered standard, any higher coursework percentage requires a specific justification. The general expectation is that a 10-point course will have an 80/20 split and include the equivalent of one 20-hour coursework assignment (although this can be split into several smaller pieces of coursework. The Practical Examination category should be used for courses with programming exams. You should not expect that during term time a student will have more than 2-4 hours to spend on a single assignment for a course per week. Please note that it is possible, and in many cases desirable, to include formative assignments which are not formally assessed but submitted for feedback, often in combination with peer assessment.]

The course will have a single coursework, which typically takes 25-hour time to finish and counts 25% towards a student's final grade. The designed coursework is to test a student's ability to apply the learned techniques to design and analyse complex algorithms, or solved practical problems occurring in practice.

Academic description:

[A more technical summary of the course aims and contents. May include terminology and technical content that might be more relevant to colleagues and administrators than to students.]

- High-Dimensional Spaces
- Best-Fit Subspaces and Singular Value Decomposition
- Spectral Graph Theory: The Cheeger Inequality, Expander Mixing Lemma
- Algorithms for Massive Data Problems: Streaming, Sketching, and Sampling: AMS and BJKST algorithm, Count-Min Sketches
- Clustering (k-means clustering, spectral clustering)
- Graph Sparsification (Cut Sparsification, Spectral Sparsification)

Syllabus:

[Provide a more detailed description of the contents of the course, e.g. a list of bullet points roughly corresponding to the topics covered in each individual lecture/tutorial/coursework. The description should not exceed 500 words but should be detailed enough to allow a student to have a good idea of what material will be covered in the course. Please keep in mind that this needs to be flexible enough to allow for minor changes from year to year without requiring new course approval each time.]

- High-Dimensional Spaces
- Best-Fit Subspaces and Singular Value Decomposition
- Spectral algorithms for massive datasets
- Data streaming algorithms
- Clustering
- Graph sparsification

Relevant QAA Computing Curriculum Sections:

[Please see http://www.qaa.ac.uk/en/Publications/Documents/SBS-Computing-consultation-15.pdf to check which section the course fits into.]

Computer Science.

Graduate Attributes, Personal and Professional skills:

[This field should be used to describe the contribution made to the development of a student's personal and professional attributes and skills as a result of studying this course – i.e. the generic and transferable skills beyond the subject of study itself. Reference in particular should be made to SCQF learning characteristics at the correct level http://www.sqa.org.uk/files_ccc/SCQF-LevelDescriptors.pdf].

As the outcome of the course, a student should be able to apply the learned mathematical knowledge to analyse and process massive datasets, and use these tolls to solve algorithmic problems occurring in practice.

Breakdown of Learning and Teaching Activities:

[Total number of lecture hours and tutorial hours, with hours for coursework assignments.]

[The breakdown of learning and teaching activities should only include contact hours with the students; everything else should be accounted for in the Directed Learning and Independent Learning hours.

The total being 10 x course credits. Assume 10 weeks of lectures slots and 10 weeks of tutorials, though not all of these need to be filled with actual contact hours. As a guideline, if a 10-pt course has 20 lecture slots in principle, around 15 of these should be filled with examinable material; the rest should be used for guest lectures, revision sessions, introductions to assignments, etc. Additional categories of learning and teaching activities are available, a full list can be found at:

http://www.euclid.ed.ac.uk/Staff/Support/User_Guides/CCAM/Teaching_Learning.htm]

Lecture Hours: _15_ hours

Seminar/Tutorial Hours: _5_ hours

Supervise practical/Workshop/Studio hours: __0__ hours

Summative assessment hours: _2__ hours

Feedback/Feedforward hours: __8__ hours

Directed Learning and Independent Learning hours: _70__ hours

Total hours: __100__ hours

You may also find the guidance on 'Total Contact Teaching Hours' and 'Examination & Assessment Information' at:
http://www.studentsystems.ed.ac.uk/Staff/Support/User_Guides/CCAM/CCAM_Information_Captured.html

Keywords:

[A list of searchable keywords.]

Algorithms, Data Science, Theoretical Computer Science

SECTION 3 - COURSE MATERIALS

3a. Sample exam question(s)

[Sample exam questions with model answers to the individual questions are required for new courses. A justification of the exam format should be provided where the suggested format non-standard. The online list of past exam papers gives an idea of what exam formats are most commonly used and which alternative formats have been http://www.inf.ed.ac.uk/teaching/exam_papers/.]

Sample Question: Let *G* be an undirected graph with conductance $\Phi_G = \Theta(1)$. Design a nearly-line time algorithm that finds a vertex set S with conductance with $\Phi_G(S) = \Theta(1)$.

Sample Answer: (1) Apply the power method to approximately compute the eigenvector associated with the 2nd smallest eigenvalue of the normalized Laplacian matrix of G; (2) Apply the algorithm behind proving the Cheeger inequality. Both steps work in time nearly linear in the size of the input graph. The approximation guarantee comes from the application of the Cheeger inequality.

3b. Sample coursework specification

[Provide a description of a possible assignment with an estimate of effort against each sub-task and a description of marking criteria.]

Possible Assignment: Given a sequence of rectangles as input in the data streaming model, in which every rectangle i is represented by its left-bottom coordinate $(x_{i,1}, y_{i,1})$ and right-top coordinate $(x_{i,2}, y_{i,2})$. Design a sublinear space algorithm that approximately computes the size of the union of all the rectangles.

Marking criteria: The high-level idea of solving this problem is to build a reduction from the described problem to the problem of estimating the number of different items in a data stream. The later problem can be solved by applying the AMS algorithm. However, different reductions will significantly influence the space complexity, and update/query time of a student's proposed algorithm. I believe that 10-hour work is needed in order to present an algorithm with non-trivial update and query time.

3c. Sample tutorial/lab sheet questions

[Provide a list of tutorial questions and answers and/or samples of lab sheets.]

None.

3d. Any other relevant materials

[Include anything else that is relevant, possibly in the form of links. If you do not want to specify a set of concrete readings for the official course descriptor, please list examples here.]

Main reference of the course:

- Avrim Blum, John Hopcroft, and Ravindran Kannan: Foundations of Data Science. https://www.cs.cornell.edu/jeh/book.pdf

SECTION 4 - COURSE MANAGEMENT

4a. Course information and publicity

[Describe what information will be provided at the start of the academic year in which format, how and where the course will be advertised, what materials will be made available online and when they will be finalised. Please note that University and School policies require that all course information is available at the start of the academic year including all teaching materials and lecture slides.]

The course webpage and reading materials will be available at the start of the academic year. Similar reading materials have been used for the Algorithmic Reading Group within the School of Informatics, which attracted many PhD students and PostDocs from different institutes to attend. These students and PostDocs could help advertise the course to students with different academic background.

4b. Feedback

[Provide details on feedback arrangements for the course. This includes when and how course feedback is solicited from the class and responded to, what feedback will be provided on assessment (coursework and exams) within what timeframe, and what opportunities students will be given to respond to feedback.

The University is committed to a baseline of principles regarding feedback that we have to implement at every level, these are described at http://www.docs.sasg.ed.ac.uk/AcademicServices/Policies/Feedback_Standards_Guiding_Principles.pdf.

Further guidance is available from http://www.enhancingfeedback.ed.ac.uk/staff.html.]

A sample solution of the coursework will be released one week after the coursework's deadline. In addition to the feedbacks of the coursework, we will provide students with solutions of the exercise questions proposed in class or listed in the main reference book. We will also provide students with 1-hour drop-in session every week to answer students' questions related the content of every week's lectures.

4c. Management of teaching delivery

[Provide details on responsibilities of each course staff member, how the lecturer will recruit, train, and supervise other course staff, what forms of communication with the class will be used, how required equipment will be procured and maintained. Include information about what support will be required for this from other parties, e.g. colleagues or the Informatics Teaching Organisation.]

- The course will be lectured by Dr. He Sun, who is a senior lecture at the School of Informatics and has worked at the intersection of Algorithm Design, and Machine Learning. He has been invited to give more than 30 conference talks internationally in this field, and is a natural candidate to deliver the course.

- He Sun will have a PhD student Bogdan Manghuic arriving in September 2018. Manghuic is familiar with most topics to be covered in the course, and has excellent communication skill. Hence, Manghuic could be acted as the TA of the course.

- The Informatics Teaching Organisation needs to arrange the lecture rooms for the course.

SECTION 5 - COMMENTS

[This section summarises comments received from relevant individuals prior to proposing the course. If you have not discussed this proposal with others please note this].

I discussed the name and topics of the proposed course with several people in the school, including Mary Cryan, Kousha Elessami, Heng Guo, Rik Sarkar.

They all think that the proposed course covers interesting and important topics that have been mainly developed in the past decade, and formed the algorithmic foundations of data science. They also think that a new course covering these state-of-the-art could be very popular among students.

5a. Year Organiser Comments

[Year Organisers are responsible for maintaining the official Year Guides for every year of study, which, among other things, provide guidance on available course choices and specialist areas. The Year Organisers of all years for which the course will be offered should be consulted on the appropriateness and relevance on the course. Issues to consider here include balance of course offerings across semesters, subject areas, and credit levels, timetabling implications, fit into the administrative structures used in delivering that year.]

5b. BoS Academic Secretary

[Any proposal has to be checked by the Secretary of the Board of Studies prior to discussion at the actual Board meeting. This is a placeholder for their comments, mainly on the formal quality of the content provided above.]