Petar Veličković
Trinity College
pv273

Project Proposal

Computer Science Tripos, Part II

Molecular multiplex network inference

23 October 2014

Project Originators:
Dr Pietro Liò
Petar Veličković

Project Supervisor:
Dr Pietro Liò

Director of Studies:
Dr Arthur C. Norman

Project Overseers:
Prof Peter Robinson
Dr Robert N. Watson

Introduction and Description of the Work

Machine learning, and statistical analysis in general, are crucial methods in the development of the field of diagnostic medicine. Determining whether or not, for example, a patient is infected with a certain type of disease given the symptoms he is experiencing or laboratory results, can be ideally presented as a problem solved by a classifier. As another example, we may be interested in knowing whether the patient is at significant risk of developing a disease, or, if he/she already is affected, the likely prognosis of the development of the disease – these may often be represented as regression problems.

The importance of such models is greater than ever, now that various kinds of biomolecular data can be extracted with greater certainty. However, most current machine learning model implementations will tend to operate on a single type of data only. This can still provide us with precise inferences, however in reality most of the data types, particularly at the molecular level, exhibit a level of *correlation* that is differently pronounced depending on the biological process/disease in question. It is therefore to be expected that taking into account several data types at once and modelling their interactions correctly within our machine learning model should provide us with even better inferences. These interactions are not fully understood, but are assumed to be more complicated than what simply combining two separate structures in a predictable way can model. As such, there is a need for creating a data structure which may be trained to learn from data sets of each individual type as well as to model their correlation.

With this project I propose a possible solution to this problem – combining multiple *hidden Markov models* (HMMs) over identical sets of nodes, each of which has been individually trained on a single type of data, with additional interlayer links – this kind of multilayered graph is known as a *multiplex network*. To the best of my knowledge, there currently exists no open-source implementation of a structure like this. As multiple types of correlated data arise in a multitude of fields, it is expected that a generic implementation of this project will prove useful not only to bioinformaticians, but essentially anyone having to perform any kind of statistical analysis. The language of choice for the implementation of this project is C++, because I already have substantial experience in it, and it is optimised for performance – hence, it can accommodate evaluation on larger data sets compared to the other options I had.

Starting Point

The project implementation will draw material from the following courses of the Computer Science Tripos:

$\sim$ **Bioinformatics** Being the central project area, this course provides an overview of biological contexts behind the machine learning models utilised, as well as examples of hidden Markov model usage in analysing biomolecular data;

$\sim$ **Artificial Intelligence I/II** The material covered in these courses is closely related to the methods utilised in this project, such as machine learning and optimisation. In particular, the standard algorithms on hidden Markov models are covered within it.

$\sim$ **Programming in C/C++** As previously mentioned, C++ will be the language of choice for the implementation of this project. I already have an extensive experience with the language, having used it primarily for competitive algorithmic programming. This course has provided me with an introduction to templates, which I will be using extensively to create as generic library elements as possible.

The Bioinformatics course is ongoing as this proposal is being written, while I will need to familiarise myself with the relevant Artificial Intelligence II material in advance. Material relevant to other aspects of the project (primarily in the form of academic papers and open-source projects) will be investigated in the initial stages, as outlined in the timetable.

Substance and Structure of the Project

As outlined in the introductory section, this project intends to produce a data structure for supervised machine learning (separate training and test data sets for purposes of classification and/or regression) which will accommodate for multiple types of data that are correlated in an undefined way with respect to the problem at hand. The key three building blocks of this data structure are as follows:

- **Hidden Markov models** (HMMs) will be used to model the solution to the problem by utilising each individual data type provided. More precisely, the full data structure will consist of multiple HMMs over the same set of nodes (for example, these may represent genes or patients), each of which has been trained on a particular type of data.

- **Multiplex networks** are then used to intertwine these individual layers together. In the most general form, a multiplex network over L layers and n

nodes can be represented as an $L \times L$ matrix of $n \times n$ matrices

$$\mathbf{M} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1L} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{L1} & \mathbf{A}_{L2} & \cdots & \mathbf{A}_{LL} \end{pmatrix}$$

where $\mathbf{A}_{ii}$ corresponds to the edge weight matrix representing the ith individual layer, that is, $(\mathbf{A}_{ii})_{jk}$ represents the weight of the edge between nodes j and k in the ith layer. Similarly, $\mathbf{A}_{xy}$ $(x \neq y)$ corresponds to the interlayer connections between layers x and y. More precisely, $(\mathbf{A}_{xy})_{ij}$ represents the weight of the edge between node i in layer x, and node j in layer y. In practice, the interlayer connections are usually modelled in the form $\mathbf{A}_{xy} = \omega_{xy}\mathbf{I}$ (where $\mathbf{I}$ is the identity matrix), i.e. a node is only linked to its own image in the other layer, and all edges between a particular pair of layers have equal weight.

A multiplex network is a special case of *multilayered graphs*, enforcing that all the layers are built over the same set of nodes. The solution to the problem of combining correlated data sets for machine learning may be pursued by considering construction of general multilayered graphs, however these fall out of the scope of the types of data used for evaluating this project.

- **Multiobjective genetic algorithms** are to be used in the final step of the construction of the data structure, which is training the multiplex – that is, determining the interlayer connection matrices $\mathbf{A}_{xy}$ (usually just the coefficients ω_{xy}, as discussed above) that will yield optimal inferences on the test set. These algorithms solve the *multiobjective optimisation problem*, which may be formulated as minimising several functions $f_1(\vec{x}), f_2(\vec{x}), \ldots, f_n(\vec{x})$ simultaneously. For the purposes of training the multiplex network, we would want to optimise the probabilities of each training set being properly classified by the model.

The execution of the project has been split into five main objectives. Their completion has been specifically marked within the milestones given in the project timetable below. The objectives are as follows, in chronological order:

- **Preparation** – this objective consists of preparatory background reading on the theory behind the data structures and algorithms given above. To confirm successful completion of this chapter as well as commence work on the full project, a working implementation of a hidden Markov model should be produced at the end of this phase and tested on data provided in the literature.

- **Core project implementation** – this objective consists of implementing the main building blocks and integrating them in a full data structure with a classification inference algorithm. Most, if not all, tests will be executed with a two-layered multiplex, but the design should be in principle extendable to an arbitrary amount of layers.

- **Core project evaluation** – this objective consists of evaluating the performance of the implementation mentioned above, both on small and large data sets. The evaluation will be performed in two main ways:

  - **Supervised learning setup**: dividing the given data sets into *training* and *testing* subsets, constructing the implemented data structure using the training set and afterwards examining the quality of its predictions on the testing set, against the known classifications provided therewith;
  - **Comparison with individual layers**: testing the gain we have made by combining different types of data. This involves constructing the individual HMM layers trained on the same data as the entire structure, using only one of the data types provided within the data sets, and then comparing the accuracy of those single-layer classifiers with the accuracy of the implemented data structure on the testing set.

  The classification problem that will be addressed in the evaluation of this project is classifying patients for diabetes. The layers of the multiplex will be trained and tested on two related types of biomolecular data that have been correlated with the presence of diabetes:

  - *DNA methylation* (amount of CH_3 (methyl-) groups attached to the base nucleotides in particular genes);
  - *Gene expression* (measure of the activity of transcription of particular genes into proteins).

- **Extensions** – this objective consists of augmenting the data structure with two additional functionalities identified as extensions: one of them is the addition of *regression* inference algorithms (producing continuous output as opposed to the discrete output of the classifier), and the other is introduction of tunable *"random noise"* in the structure (in the form of adding/removing nodes/edges, or modifying edge weights by a random amount), to accommodate for the slightly stochastic nature of these processes. Both extensions should be implemented and evaluated, time permitting, upon completion of the core project.

- **Dissertation** – this objective consists of writing up the project dissertation, documenting how all of the above objectives have been achieved in a clear and logical order.

Success Criteria

The project will be considered a success upon satisfactory completion of the **Preparation**, **Core project implementation**, **Core project evaluation** and **Dissertation** objectives as outlined above. Precisely:

- The complete proposed classifier data structure should be implemented, incorporating at least the three main building blocks outlined in the previous section. Correct operation of the individual modules within the structure should be tested on the sample tests provided in relevant literature or academic papers.
- The accuracy of the classifier should be evaluated with at least the two methods given in the previous section (supervised learning setup to estimate accuracy, and comparison with single-layered classifiers). It might prove useful to provide further ways of evaluation, e.g. of the structure's running time/space efficiency on the training and testing sets; these methods should be investigated as needed.
- Finally, a dissertation should be produced, documenting the necessary introduction to the problem area, the work done in the stages of preparation, implementation and evaluation of the project, and the conclusions drawn from the project's overall execution.

The **Extensions** objective should also be achievable, however it is orthogonal to the core project and as such is not essential to its success.

Resources Required

I intend to use my own laptop (2.6 GHz Intel Core i7 with 8 GB RAM, running Mac OS X 10.10 Yosemite) for the purposes of implementation, evaluation and dissertation writeup. I accept full responsibility for this machine and I have made the following contingency plans to protect myself against hardware/software failures:

- Revision control of the project's codebase and dissertation using `git`, with all commits pushed onto a remote private repository on GitHub (with intents of making it public upon completion of the project);
- Synchronisation of the entire project with my personal file spaces on Dropbox, Google Drive and the MCS utilising `rsync`, both manually and automatically;
- Regular backups of the machine's entire filesystem (and hence the project files as well) utilising Apple's Time Machine, on an external 1TB HDD.

For evaluation purposes, the project will also utilise molecular data (DNA methylation and gene expression matrices of patients, with provided classifications for diabetes) located in the Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/). This data is readily available, and I am familiar with the methods of accessing the relevant sets.

This project is not expected to rely on any libraries for the three major modules outlined in the project substance, as some proposed features such as adding random noise to the model and making the model generic may not be easily integrated with the current implementations available, justifying an implementation from scratch. However, in the event of a major schedule overrun, the project implementation may fall back to extending an existing library.

Timetable and Milestones

To facilitate execution of the project in accordance with the proposed structure, I have divided the project development timetable into fifteen fortnightly slots, some of which have been reserved for initial research and dissertation writeup, and some serving as buffer time in the event of schedule overruns. To keep track of the progress of the project, I have assigned a milestone to each slot – it should be verified at the end of each slot that its respective milestone has been achieved.

The rough plan is to complete the core project by the end of December, implement proposed extensions by mid-February, and produce a final dissertation writeup ready for submission by late April. The full proposed timetable is given below:

Slot 0: *6 October – 22 October*

→ Working on the project proposal.

→ Setup of all contingency schemes as described in the resource declaration section.

→ Studying the essentials of hidden Markov models from relevant literature and the previous year's Artificial Intelligence II course notes.

Milestone: Project proposal ready for submission.

Slot 1: *23 October – 5 November*
DEADLINE: Project proposal submission (24 October)

→ Researching academic papers relevant to multiplex networks, and investigating genetic algorithm implementations and their relative merits.

→ Starting work on a generic HMM implementation in C++.

Milestone: A basic working HMM implementation, tested on a few examples given in literature (**Preparation completed**).

Slot 2: *6 November – 19 November*

→ Implementation of a generic multiplex network class.

→ Integration of the HMM as a layer in the multiplex.

Milestone: Successfully connecting two HMMs in a multiplex network. Providing a front-end command line tool which can read and analyse a multiplex given in a prescribed format.

Slot 3: *20 November – 3 December*

→ Implementation of inference algorithms on multiplex networks; verification that the algorithms produce expected outputs on known networks.

Milestone: Inferences of comparable quality successfully made on known examples of multiplex networks (from research papers).

Slot 4: *4 December – 17 December*

→ Implementation of a suitable genetic algorithm to be used for training the interlayer edge weights for the multiplex network.

Milestone: Working generic implementation of the genetic algorithm, tested on example functions given in relevant academic papers.

Slot 5: *18 December – 31 December*

→ Integration of the genetic algorithm with what was previously done; utilising it to train the underlying multiplex as described.

→ Attempting a first evaluation run on a small data set; fixing any residual bugs in the design as found.

Milestone: Classifier successfully ran on a small data set (**Core project implementation completed**).

Slot 6: *1 January – 14 January*

→ Evaluating the core project by training and testing on large data sets; in particular, performing a comparison with HMM classifiers over a single data type.

→ Writing the progress report; if necessary, sending it to the supervisor and incorporating any comments.

Milestone: Progress report ready for submission. Classifier performance evaluated on large data sets (**Core project evaluation completed**).

Slot 7: *15 January – 28 January*

→ Implementing a regression method of inference on the multiplex HMM structure. Running on the provided data sets, and commenting on the obtained results.

→ Prepare slides for the progress report presentation (utilising output from the evaluation as needed).

Milestone: Successful regression performed on the multiplex, producing useful output for the given training and testing data sets. Presentation slides prepared.

Slot 8: *29 January – 11 February*
Deadline: Progress report submission (30 January)

→ Rehearse and deliver presentation to overseeing group.

→ Implementation of a feature to allow introducing random noise into the multiplex structure, either by adding extra nodes/edges or modifying the values of existing ones. Commenting on performance with respect to the noise parameter(s).

Milestone: Presentation successfully delivered. A working scheme for adding noise into the model, with appropriate testing conducted (**Extensions completed**).

Slot 9: *12 February – 25 February*

→ Buffer slot #1, to accommodate any schedule overruns with the implementation; if there are none, start work on the dissertation early.

Milestone: None.

Slot 10: *26 February – 11 March*

→ Starting work on the dissertation; completion of preparatory chapters.

Milestone: Completion of the Introduction and Preparation chapters of the dissertation.

Slot 11: *12 March – 25 March*

→ Completion of draft dissertation.

→ Submission of the draft to the supervisor and Director of Studies.

Milestone: Completion of the Implementation, Evaluation and Conclusion chapters of the dissertation. Submission of the first draft.

Slot 12: *26 March – 8 April*

→ Receipt of responses from the supervisor, Director of Studies and any proofreaders.

→ Incorporation of the obtained comments as appropriate, in preparation of the second draft of the dissertation.

Milestone: Submission of the second draft of the dissertation to the supervisor and Director of Studies.

Slot 13: *9 April – 22 April*

→ Receipt of responses on the second draft.

→ Incorporate any further comments as appropriate.

Milestone: Final version of the dissertation ready for submission.

Slot 14: *23 April – 6 May*

→ Print, bind and submit the dissertation.

Milestone: Submission of dissertation (**Dissertation completed**).

Slot 15: *7 May – 13 May*

→ Buffer slot #2, to accommodate any unexpected issues with the writeup of the dissertation.

→ Revision for Part II examinations.

Deadline: Dissertation submission (15 May)