COMP9417 17s1 Assignment 2 - project topics

Assignment criteria.

Brief guide to writing the report.

Sample of a report.

Here are the topics (more may become available, so ask me if there is nothing of interest here or you don't want to develop your own project):

Topic 0: Propose your own topic

Task description
Up to you. The basic conditions are:

- it must involve some practical work with some implementation of machine learning
- you must send me an email with a description of what you are planning (a couple of paragraphs would usually be enough) that I need to approve before you start
- it must not involve double-dipping, i.e., be part of project for another course, or for research postgrads it must include a statement to the effect that it is not part of the main work planned for the thesis (although it can be related, e.g. if your research is on behavioural cloning, it could be on reinforcement learning, which is a related but different approach)

Team 1 to 4 person team

Difficulty assigned per topic (often 5/5).

Topic 1: Machine learning and data mining competitions

Task description
A number of sites now host regular competitions. You can enter a live competition or work on the dataset from a past competition. The main site hosting machine learning competitions is Kaggle. Current competitions are here. Two others that appear currently not very active are TunedIT and MLcomp, but they might have interesting datasets for you.

Some of the past competitions could be of interest, e.g., the Million Song Dataset Challenge. See also Topic 8 below. The choice of competition is up to you. The basic conditions are:

- assess carefully the time you will need to understand the competition requirements, get familiar with the data and run the algorithm(s) you plan to use
- you must send me an email with a description of what you are planning (a couple of paragraphs would usually be enough) that I need to approve before you start
- your report and software must be submitted as for the other topics; for live competitions you can include your submission's placing on the leaderboard at submission time!

Team 1 to 4 person team

Difficulty assigned per topic (often 5/5).

Topic 2: Data repositories

Task description
As an alternative to following the rules of past competitions you can choose to apply machine learning to the

datasets in new and interesting ways, or you could do the same with datasets from some of the many repositories now on the Web. The original such repository is the UC Irvine Machine Learning Repository here. These datasets are usually formatted for easy import into machine learning software.

MIT's Reality Commons project has some datasets on human behaviour obtained from mobile phones and other devices, such as the one on Human Activity Recognition Using Smartphones here.

The U.S. Government's open data initiative Data.gov contains many data sets. The problem is that many are simply raw data and individual data sets do not represent a well-defined learning problem; however, the hope is that by linking several data sets some interesting mining may be possible. Other governments doing similar things are in the U.K., N.Z and Australia. A more complete list is available here.

Many of the above, plus more, are included in the list of repositories at KDnuggets.

Pick a data set, review its publication history (if any) and see if you can think of an interesting angle to explore to make a project. The basic conditions are

- assess carefully the time you will need to specify a well-posed problem and set up the dataset(s) and algorithm(s)
- give particular attention to defining a target function and ways in which you can evaluate the learning
- you must send me an email with a description of what you are planning (a couple of paragraphs would usually be enough) that I need to approve before you start

Team 1 to 4 person team

Difficulty assigned per topic (often 5/5).

Topic 3: Perceptron / Linear Unit

Task description
Write a program to implement learning for numeric prediction / classification based on an *n*-input linear unit / perceptron architecture. First, implement the linear unit architecture and the gradient descent learning method. Test your system on the standard UCI data set autoMpg. Note that you will have to think about the representation of input data for this task: some of the attributes are real-valued, but others have discrete, ordered values. One possibility is to use a Boolean variable (an "indicator" variable) for each discrete value for such attributes, where the value is 1 if the instance has that value for the attribute, and 0 otherwise.

Second, implement the perceptron training rule to learn *n*-input perceptron classifiers for Boolean functions. Test on examples from at least two different **8-input** Boolean functions (note: should include linearly-separable and non-linearly-separable functions).

There is an option in this project to add graphical output showing how the system is learning. This requires adding an extra person to the team (from 2 to 3 people). The graphics do not have to be very complicated, but should show key features of the learning algorithms, such as weight update.

Team 2 or 3 person team

Difficulty 3/5

Topic 4: Nearest Neighbour

Task description
Implement *k*-Nearest Neighbour (kNN) for both classification and numeric prediction.

For classification, test your version of kNN for range of values of k on the standard UCI data set ionosphere. Evaluate your system by leave-one-out cross-validation.

For numeric prediction, test your version of kNN for range of values of k on the standard UCI data set autos. For this data set you can remove examples with missing attribute values, which will reduce the number of examples to 159. The task is to predict the price given the 14 continuous and 1 integer attributes. You can also experiment with encoding the other attributes to allow their use in the distance function to see if this affects predictive accuracy. Evaluate your system by leave-one-out cross-validation.

If time permits you can extend your methods to implement distance-weighted Nearest Neighbour (WNN).

There is an option in this project to add graphical output showing what the system "learns". This requires adding an extra person to the team (from 2 to 3 people). The graphics does not have to be very complicated, but needs to show key features of the algorithms, such as the effect of varying the *k* parameter.

Team 2 or 3 person team

Difficulty 3/5

Topic 5: Genetic Algorithm

Task description
Implement the basic Genetic Algorithm method as described here, including the point mutation rule and at least two of these crossover rules - single-point, two-point and uniform. Test on the standard UCI data sets balance-scale and mushroom.

You will need to consider carefully the representations you use for the classifiers to be learned for both data sets. Also note that the mushroom data set is quite large.

There is an option in this project to add graphical output showing how the system is learning. This requires adding an extra person to the team (from 2 to 3 people). The graphics does not have to be very complicated, but needs to show key features of the algorithm, for example, the selection process to construct a new generation of hypotheses.

Team 2 or 3 person team

Difficulty 3/5

Topic 6: Sequential Covering

Task description
Implement a Sequential Covering rule learning algorithm. Base it on the FOIL algorithm in Table 10.4, p. 286 of Mitchell's book but use an attribute-value constraints representation for the rules, i.e., propositional logic, as in Table 10.2, p. 278 of Mitchell's book. You will also need to implement a version of information gain for this representation, rather than the first-order version used in FOIL, or an alternative of your own.

Test on these versions of the UCI data sets mushroom and splice.

Note that you do not need the instance name (first attribute) in the splice data set, and that the mushroom data set is quite large !

There is an option in this project to add graphical output showing how the system is learning. This requires adding an extra person to the team (from 1 to 2 people). The graphics does not have to be very complicated, but needs to show key features of the algorithm, for example, the coverage and accuracy of the process of learning a rule.

Team 1 or 2 person team

Difficulty 3/5

Topic 7: Inductive Logic Programming

Task description
Implement an algorithm to compute the Relative Least General Generalization (RLGG) of pairs of ground instances of the target predicate, as outlined in the lecture notes.

Data set: In a typical Bongard problem there are 6 positive and 6 negative examples of the hidden rule that correctly classifies a scene. Select a small number (at least 2) of the Bongard problems from here (you could also make your own, or contact me if you need more information).

You will need to represent the scenes in such a way that the RLGG can be represented as a single clause. The RLGG is constructed from a pair of positive examples, so in a typical Bongard problem there would be 15 possible pairs to choose to initialize RLGG construction.

Once constructed, the RLGG clause (which is unique for a pair of positive examples and fixed set of background clauses) can be further generalized to ensure it covers all the positive examples and none of the negative examples by dropping literals or turning constants to variables.

Represent the examples (pos and neg) and the background knowledge in Prolog. It may be easiest to implement the algorithm in Prolog too. There are several high-quality open-source Prologs. SWI Prolog has the best documentation, with built-in help, whereas in general YAP Prolog enables faster running compiled programs.

Team 1 or 2 person team

Difficulty 4/5

Topic 8: Work on a past KDD Cup competition task

Task description
The ACM KDD Cup competitions have been running annually since 1997. These are challenging machine learning tasks, and are chosen to be representative of problems faced in industry and academia. However, note that in these competitions it is often the case that, as in many real-world data mining applications, it is usually as important to get to know the data, deal with issues in setting up the problem, etc. as it is to work out how to actually do the learning.

Data set: select **one** of the following tasks from past competitions:

Year	Task description
1999	Network intrusion detection as a classification task
2002	Text-based categorization of gene activity in yeast - Task 2
2003	Predicting paper citations in the arXiv network - Task 1
2004	Scientific prediction tasks in protein homology or particle physics
2009	Prediction tasks in a telecommunications company, such as churn, appetency or up-sell - Use small dataset

You must follow the task description as closely as possible. Since the competitions have closed, the test data are available, but you should avoid testing against these until you are really sure you understand the performance of your method(s). Test set performance should be evaluated and appear in your report, as well as an indication of how your team's performance matches those in the original competition. cases,

Team 1 to 4 person team

Difficulty 5/5

Topic 9: Recommender system using collaborative filtering

Task description
Pick **one** of the the following problems for recommendation and apply collaborative filtering (or an alternative). The data is from a collection collected by the GroupLens research group.

The suggested problems are either: learning to predict the ratings of books on the BookCrossing dataset, or movie ratings on the MovieLens Data Sets (these are of different sizes, so start with the 100K). However, other options are also available from the GroupLens site.

Team 1 to 4 person team

Difficulty 5/5

Topic 10: Neural networks

Task description
This is Mitchell's face recognition task using neural networks. The task is explained here. See also Chapter 4 in Mitchell's book.

Note that although the task description is clearly dated, the problem remains an interesting one. You need to complete the second part of the task as well (ignore the ``optional'' heading !). This is an open-ended assignment, so you will need to consider how far you can take this this.

Team 2 or 3 person team

Difficulty 3/5

Topic 11: Text categorization

Task description
Implement a version of the Naive Bayes classifier and apply it to the text documents in the 20 newgroups data set. Adopt the same approach as described in Chapter 6 of Mitchell's book. The data set is available here.

Team 1 person team

Difficulty 2/5

Topic 12: Reinforcement Learning

Task description
Implement a traffic light simulator for a single intersection and a reinforcement learning system to learn how to switch the traffic lights to reduce the delay for vehicles at the intersection.

The details are explained here and there is a short movie showing the traffic light controller in operation here.

Implementing graphics will help considerably in debugging this project.

Team 3 or 4 person team

Difficulty 4/5 to 5/5 depending on extensions.

Topic 13: Reinforcement Learning - BOXES pole-balancing challenge

Task description
BOXES was an early reinforcement learning algorithm which is easy to implement and still gives quite

reasonable performance on some benchmark problems such as the pole-and-cart. The challenge is to implement a reinforcement learning method that beats BOXES on the pole-and-cart. One candidate would be to try Q-learning.

BOXES is introduced here with some sample code in Java. A version of the pole-and-cart simulator you will need is also downloadable from the same location. I can probably also provide a version of the simulator in C if you prefer.

Implementing graphics will help considerably in debugging this project.

Team 3 or 4 person team

Difficulty 4/5

Topic 14: Bioinformatics

Task description
Since these can depend on domain knowledge you will need to contact me for details.

Team 1 to 4 person team

Difficulty assigned per topic (probably 5/5).

Topic 15: Learning to play "20 Questions"

Task description
The traditional game often called "20 Questions" has been implemented in at least two online versions that learn from user interaction, 20Q and Akinator. It is not clear how the learning is implemented for either of these systems, but there appears to be a patent application on the technology behind 20Q by its author, Robin Burgener, which outlines a form of neural network. In this project, which is very open-ended, the objective is to implement a simple version of the twenty questions game that can interact with users and learn which questions to ask, thereby (hopefully) improving with experience. Some general background and information about radio and TV versions is available on Wikipedia. There are various pointers on the web to different implementations, for example, in Python, or suggestions about the way in which this kind of system could use machine learning, but these should be treated with caution. This project is only for those with experience in the rapid development of interactive systems who are looking for a challenging application of machine learning in this context.

Team 1 to 4 person team

Difficulty 5/5

Topic 16: (Probabilistic Modelling) Driver Telematics Analytics

Task description
This topic concerns the task of probabilistic unsupervised classification for *driver telematic analytics*. It is a previous Kaggle competition and the main goal is to identify a driver signature using telematic data. The link to the competition details is here. You are required to:

1. Familiarize yourself with the problem and the data
2. Propose a solution using concepts and techniques explained during the lecture (in particular, the sessions on probabilistic graphical models). You can develop your own model but it needs to be approved by the lecturer first.
3. Propose a model solution and write up the code to execute it. You can use existing code such as the BRML toolbox, as long as you explain your solution thoroughly and give the corresponding acknowledgements.

4. Make an actual submission to the competition and report your username and position on the Leaderboard
5. Follow guidelines described above when writing your report

Team 1 to 2 person team

Difficulty 4/5

Last modified Thursday 25 May 09:25:07 AEST 2017