CS 294-34 Homework 5

April 15, 2008

1. You will experiment with the following Reinforcement Learning demos:

   Black Jack: `http://lslwww.epfl.ch/~anperez/BlackJack/classes/RLJavaBJ.html`

   Cat and mouse: `http://www.cse.unsw.edu.au/~cs9417ml/RL1/applet.html`

   Robot: `http://www.applied-mathematics.net/qlearning/qlearning.html`

   (a) Define the state-space, action-space, and rewards for Black Jack, Cat and Mouse, and Robot. How would you represent the Q function to speed up training?

   (b) Black Jack, Cat and Mouse, and Robot applets use a TD algorithm (SARSA or Q-learning). Is it possible to implement a Dynamic Programming algorithm for these tasks? What would be the pros and cons of implementing DP or a Monte Carlo algorithm?

   (c) In Cat and Mouse you can try both SARSA and Q-learning; which one works better and why?

   (d) In Cat and Mouse, how would you design the features such that the optimal policy generalizes to other environments?

   (e) Black Jack, Cat and Mouse, and Robot applets all have the following three parameters: i) alpha: learning rate, ii) gamma: discount factor, and iii) epsilon: greediness of action selection. Experiment with various values of these parameters; try the extreme values as well. What effect do these parameters have on performance of the algorithm? What are the optimal values? Is it beneficial to change their values during training?

2. For the collaborative filtering part of the homework, download the 100,000 ratings dataset from Movielens. It can be found at `http://www.grouplens.org/node/73` This is a small dataset of movie ratings that is often used to compare collaborative filtering algorithms. The 100k ratings subset will be used to limit the time that computations will require.

   The archive contains some splits of the data. For example, ua.base contains a training portion of the ratings, and ua.test contains the remaining ratings. ua is a different split. For the purpose of this exercise, use the 'ua' split.

(a) Load the data into a matrix, filling the missing ratings with a placeholder or NA. Examine the data to try to get a feel for it. Specifically, look at the proportion of data that is missing, the distribution of the ratings, how different the mean ratings for different movies are, etc.

(b) Implement the weighted-SVD collaborative filtering algorithm, as described in class. The initial matrix should be composed of the ratings in ua.base but centered (each rating should have that movie's mean subtracted off). Evaluate the performance based on the ua.test file, using the mean-squared error and mean-absolute-error metrics. Experiment with different numbers of dimensions and see how performance is affected? What is the problem with using a large number of dimensions?

(c) Can you find anything interpretable in the model? Do any of the movie eigenvectors appear to correspond to an intuative clusters? Remember that you have movie titles available. Investigate whether there is a relationship between the user eigenvectors and user demographic features (also available). Remember that this kind of analysis can be done for any matrix-factorization based method.

(d) (Optional) Implement an alternative collaborative filtering algorithm, such as Naive-Bayes or k-nearest-neighbours and compare performance. Naive Bayes will probably require smoothing to compute reasonable estimates.