4F13 Machine Learning: Coursework #2: Probabilistic Ranking

Carl Edward Rasmussen

Due: 17:00, Nov 16th, 2018 online via moodle

Your answers should contain an explanation of what you do, and 2-4 central commands to achieve it (but complete listings are unnecessary). You must also give an *interpretation* of what the numerical values and graphs you provide *mean* – why are the results the way they are? **Each question should be labelled and answered separately and distinctly.** Total combined length of answers must not exceed 1000 words; clearly indicate the actual total number of words in your coursework. All questions carry approximately equal weight.

In this assignment, you'll be using the (binary) results of the 2011 ATP men's tennis singles for 107 players in a total of 1801 games (which these players played against each other in the 2011 season), to compute probabilistic rankings of the skills of these players.

The match data is provided in the file `tennis_data.mat`, which contains two matrices: `W` is a cell array, whose `i`'th entry is the name of player `i`, and `G` is a 1801 by 2 matrix of the played games, one row per game: the first column is the identity of the player who won the game, and the second column contains the identity of the player who lost. Note, that this convention means that variable $\mathtt{y}_\mathtt{g}$ (the game outcome) in the lecture notes is always +1, and can consequently be ignored. Some rows will appear more than once (corresponding to two players having played each other several times with the same outcome).

a) Complete the code in `gibbsrank.m`, by adding the lines required to sample from the conditional distributions needed for Gibbs sampling for the ranking model discussed in the lectures. Run the Gibbs sampler, eg for 1100 iterations. Plot some of the sampled player skills as a function of the Gibbs iteration. Does it look as though the Gibbs sampler is able to move around the whole posterior distribution? What are the burn-in and mixing times for the Gibbs sampler? It may be helpful to look at the auto covariance coefficient, which can be calculated by the `xcov(samples,100,'coeff')` command.

b) Do inference in the model instead, by running message passing and EP using `eprank.m`. Explain the concept of *convergence* for both the Gibbs sampler and the message passing algorithms. What type of object are we converging to in the two cases, and how do you judge convergence, how many iterations are necessary? Do you reach the same answers independent of initialization and pseudo random number seeds?

c) For the message passing algorithm, compute two 4 by 4 tables of probabilities, including only the 4 top players according to the ATP ranking in the lecture notes. First table for the probabilities that the skill of one player is higher than the other, and second table for the probability of one player winning a match between the two. Explain the difference. The Φ function is implemented in matlab as `normcdf`.

d) For the Gibbs sampler, two ways of comparing skills are: 1) first compute the marginal skills for each player, then work out the probability that player one has a better skill, vs 2) use the joint samples from the Gibbs sampler to directly infer that probability. Examine these two alternatives on player 1 and 16. Which method is best? Using that method, derive a 4 by 4 table for the skills (not the game outcomes) and compare to that of the message passing algorithm (from question c)).

e) Compare the rankings based on outcomes for three different cases: 1) empirical game outcome averages, 2) predictions based on Gibbs sampling and 3) predictions based on the message passing algorithm. You may find the bar plot in `cw2.m` useful. Explain the differences.