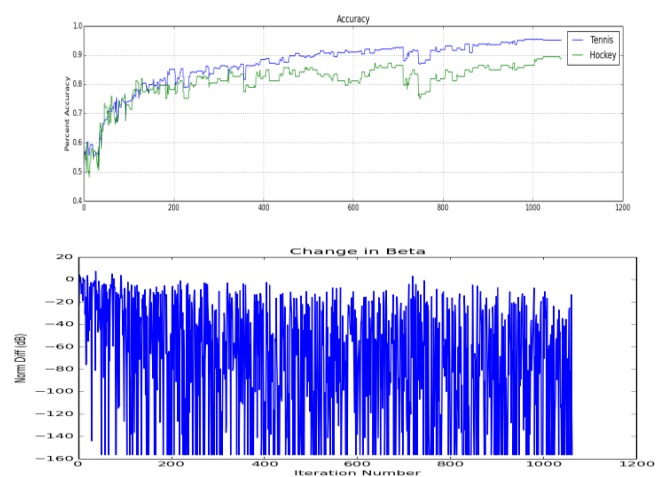
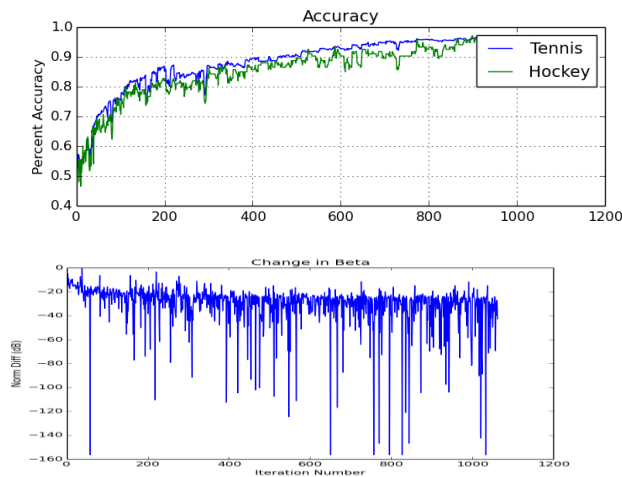


See logreg.py for detailed comments explaining the code functionality. The code will not pass the tests.py unless the variable step size is commented out in sg_update.

1. The Learning rate determines how well the optimum is found. A step size of 1 is relatively large and does not result in good performance as the steps are too coarse to settle on the minima. A step size of 0.1 or less was typically used for the tests. A variable step-size based on iteration number, initial step size, and a tuning parameter was implemented and had improved Beta convergence performance and similar prediction accuracy. Generally 0.1 fixed worked well. Tennis equals baseball in the figures. The left figures show fixed step size of 0.1 unregulated, while the right figures are for variable step. Lower plots show change in the norm of the Beta vectors from step to step in dB.



2. From the Beta convergence plots, with no regulation and a fixed step size, the Betas don't truly converge, but they change less over time. Around 600 iterations the accuracy reaches close to 90% for both document types and the change in Beta falls below -20 dB so, 600 iterations is probably a sufficient number of iterations. Variable step sizes result in smaller changes in Beta from step to step, but the changes seem to jump between nearly 0 to 1 or 2 points very rapidly.

3. As the fixed step, non-regulated case gave the best results, it's final Beta values were used to assess the best feature predictors. The most positive Beta values were used to assess good features for baseball while the most negative Beta values were used to determine good features for hockey. The top five for each were:

Baseball: hit, runs, baseball, pitching, bat

Hockey: hockey, playoffs, goals, next, playoff

These make a lot of sense. Interestingly, "hit" could go both ways and I was somewhat surprised "next" was good for hockey. I'm not clear why that was so highly rated.

4. The same Beta results were used and low absolute values were found to determine what words were not good predictors. Predictors for the two classes were simply clumped:

Poor Features: tone, blasted, memoriam, pitiful, bloody, everywhere, deceased, broad, silence, racist

Again, blasted or bloody I'd imagine could be applied to hockey. Otherwise, the poor features make sense.

I'll conclude by commenting, that while I'm confident it worked based on the test cases and debug printouts showing the last-update values were being calculated correctly, the unregulated case never performed as well as the regulated case. It did result in a consistent change in Beta norm, but would periodically spike sharply like the jump in 300 that corresponds to a dip in accuracy around iteration 300 as well. The mechanism for this anomaly is not well understood.

