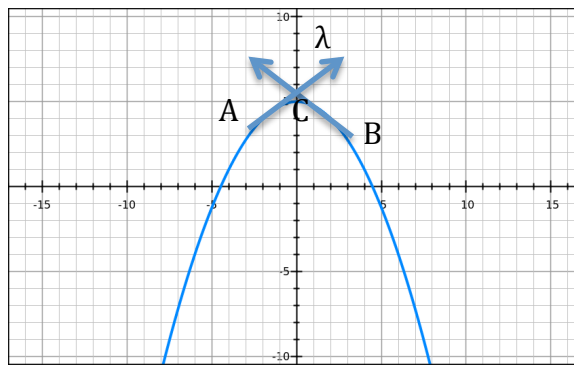


1) What is the role of the learning rate?

- The learning rate can have a major affect on the time to convergence. For example, if the learning rate is too small, it may take too long to converge to the maximized beta values. On the other hand, if the learning rate is too large, the algorithm may “bounce” over the true value forever. This can be seen in the example below. In this example, we are searching for the true value of beta C. If a step size of  $\lambda$  is taken from point A, the new beta value would be B and would create an infinite loop back and forth between A and B.



2) How many passes over the data do you need to complete?

- To achieve stable accuracy, 3 passes over the training data were required. This achieved ~99% accuracy over the training data and ~94% accuracy over the test set. More passes over the training set only fluctuated the accuracy around these values.
- 3) What words are the best predictors of each class? How (mathematically) did you find them?
- The 10 best predictors of each class (in order) were the terms “hockey”, “playoffs”, “hit”, “runs”, “pick”, “bat”, “playoff”, “points”, “saves”, and “period.” To compute these values, I took the absolute value of each final beta value and then sorted the list. I then found the indices of the max 10 values and found the corresponding vocab word for each.
- 4) What words are the poorest predictors of each class? How (mathematically) did you find them?
- The 10 worst predictors of a class were the terms “wrestling”, “vintage”, “tone”, “silence”, “rode”, “riel”, “racist”, “pitiful”, “memoriam”, and “hooked.” To find these terms, I followed the same logic as for finding the best predictors except for instead of taking the largest beta values, I took the smallest magnitude values. These beta values were all 0, so the above list is not in worst to 10<sup>th</sup> worst order. The code to solve both questions 3 and 4 is commented out at the bottom of the main function in logreg.py.