1 Solution

1. Read File
   (a) bag-of-words: use regex to retrieve all the data and store as term frequency vector
   (b) bigram: just like bag of words but put two word as a symbol instead.
   (c) store the file in the container as pattern:[dictionary(key,frequency),sentiment]
2. Shuffle
   (a) use a seed to reproduce the result
   (b) cut the train and validate data into two part
   (c) shuffle the train data
3. Train
   (a) update the weight if the prediction is wrong
   (b) only update the word occur in a single document(sparse vector)
   (c) average the weight while updating weight to enhance the run-time and reduce memory usage
4. Evaluate
   (a) use the fit function to predict the classification,return 1 as positive,and -1 as negative
   (b) sum the total of correct classifications to compute the accuracy
   (c) the top10 function sort all the weight in order to find the most positive words
   (d) use the progress function to plot the learning progress

2 Result

1. Accuracy
   (a) The starting precision will be 0.5 because the weight will not update before shuffling.However, all the result will be all positive or negative.
   (b) After shuffling the document, the weight will update and the precision will comes to around 0.5.
   (c) Multiple passes over the training instances will improve the accuracy of prediction. Instead of using the last time weight, I average the weight of each train and iterate for 20 time. The precision will stop at around 0.8375 and stop training. See Figure1.

(d) With another feature: Bigram, compare to bag-of-words, bigram put two words as one symbol. The initial accuracy ranked up to 0.67 which is much more higher than bad-of-words. However, I don't think it really help improve the accuracy because the training stop at accuracy of 0.8025, but it only took 5 time iteration to meet the result with a very high initial accuracy.

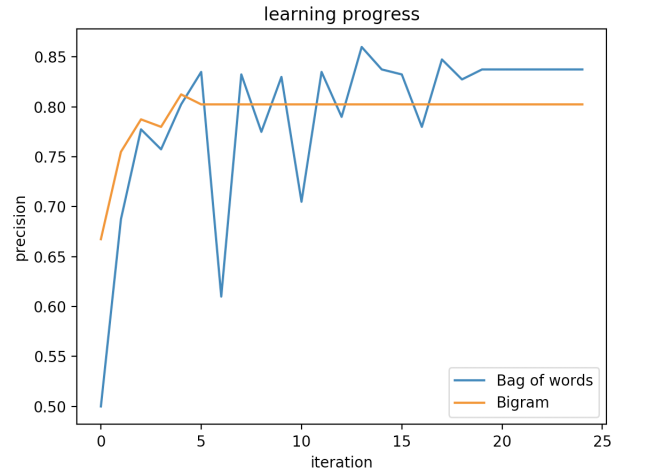

Figure 1: Learning Progress

2. The top 10 most positive word

   (a) bag-of-words:

      i. seen:247,great:230,both:188,best:180,quite:165,see:159,life:159,back,156,ever:156,most,150

   (b) bigram:

      i. black cauldron:31,he is:29, ,which is:28,the best:28,one of:27,and it:26,to the:25,the black:25,to see :24,a little:23

   (c) Better feature:

      i. As for laptop and restanrant: some symbol may be very important like battery assumption, distance(distant) that will be considered as an important factor. We can train a seperate weight with this value to make then become more influential. Just to said that we can pre-make a weight rather then set them to 0.

   (d) run time

      i. bag of words:6.36

      ii. bigram:11.9