Logistic Regression and Stochastic Gradient Ascent Analysis

Paul Laliberte' | CSCI-5622

1. How did the learning rate affect the convergence of your SGA implementation?

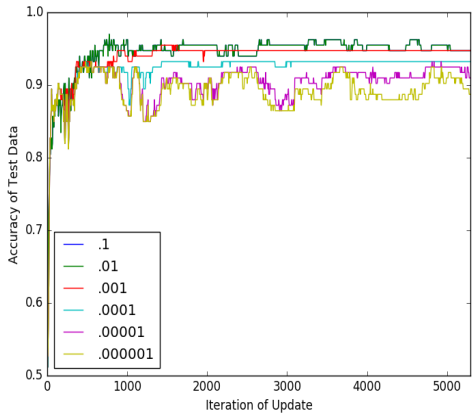

Figure 1: Learn Rate and Convergence

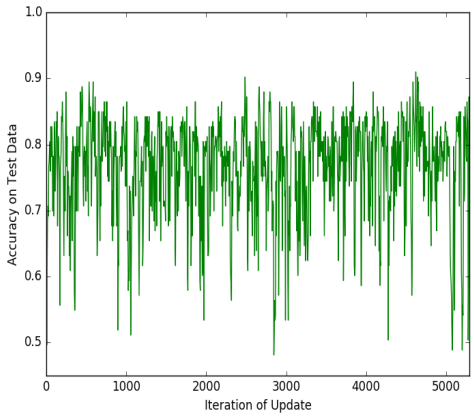

Figure 2: Lack of Converging Solution

The initial learning rate was $\eta = 0.1$, $\lambda = .00001$, and passes $= 5$ (tf-idf turned off). We decreased the rate by magnitudes of .1, i.e. .01, .001, .0001, and so on. There are two important aspects to take from Figure 1. The first being that a low initial learning rate can be favorable, in a sense, because there is minimal overfitting to the data set. However, a learning rate that is low may not be feasible, since finding a converging solution may take more time than available ($\eta = .00001$ and $\eta = .000001$ show proof of this slow convergence). Next, we consider a high initial learning rate. From Figure 1, we can infer that a higher learning rate can lead to a much faster convergence. The drawback to this faster rate is that we may be overfitting our data. A $\eta = .1$ achieves almost full convergence in only one pass through the training data. We can try to correct for this by regularization (increasing λ), but too much regularization can also cause problems (see problem 2).

2. What was your stopping criterion and how many passes over the data did you need before stopping?

The stopping criterion considered two factors: 1) Was convergence a viable possibility? 2) What was the rate of convergence? We first analyzed the issue of whether convergence was possible. To give an example, running the program with a $\eta = .1$ and $\lambda = .25$ produced a divergent solution, no matter the number of passes through the data we assigned (Figure 2). The variance over the data was consistent for both training data (TP, TA) and testing data (HP, HA), where TP and HP are the log probabilities. If there seemed to be a definitive convergence, we then considered the rate of that convergence. To be more specific, running the program with a $\eta = .001$, $\lambda = .25$, and passes $= 10$, would eventually result in a solution that was at, or near, convergence within 6 to 7 passes through the data. At this point we were getting minimal improvement (and no diminishment) in the convergence rate of TP, HP, TA, and HA. Hence, we could stop and be confident in the results. In summary, if there occurred a point where one pass through the data only resulted in minimal improvement of convergence, and did not diminish, then we considered this an acceptable stopping point.

3. What words are the best predictors of each class? How (mathematically) did you find them?

To classify words as good predictors we used the tf-idf weight with $\eta = .1$, $\lambda = 0.00001$, and passes $= 100$. We define

$$\text{tf}(t) = \frac{\#\text{ of times term } t \text{ appears in doc}}{\text{total }\#\text{ of term } t \text{ in doc}},$$

and

$$\text{idf}(t) = \log\left(\frac{\text{total }\#\text{ of docs}}{\#\text{ of docs with term } t}\right).$$

Furthermore, the tf-idf weight is given as

$$\text{tf-idf}(t) = \text{tf}(t) \times \text{idf}(t).$$

We then summed every tf-idf calculation as we passed through the data. The highest sum of tf-idf weighted scores were the best predictor words. We also took into consideration words that did not appear in the relevant classes, and removed them from the rating (assigned a special value to them). The twenty best predictor words of each class can be found in the first two tables, from the left, in Figure 3 (auto and then cycle). We can see that there are several words that overlap between the two, but there are also very distinct words that we would assume represent best predictors of each class (car, cars, bike, ride, riding).

4. What words are the poorest predictors of classes? How (mathematically) did you find them?

To classify words as bad predictors, we used the same sum of tf-idf weights that were calculated in (3). The lowest sum of tf-idf weighted scores were the worst predictor words. The twenty worst predictor words of each class can be found in the last two tables, from the left, in Figure 3 (auto and then cycle).

	term	tfidf (sum)
0	car	99.304331
1	distribution	72.543926
2	like	61.255098
3	one	58.244466
4	usa	54.896685
5	cars	49.358891
6	reply	48.777984
7	know	48.750605
8	good	47.470158
9	new	46.013241
10	get	45.351627
11	please	42.591214
12	thanks	41.668738
13	also	38.686989
14	think	38.008484
15	time	36.448811
16	ca	32.819943
17	much	32.645049
18	right	30.715554
19	want	30.276095

	term	tfidf (sum)
0	dod	100.652430
1	bike	85.889095
2	one	68.474187
3	like	67.807797
4	distribution	55.874916
5	get	49.700894
6	know	47.163276
7	ca	47.049867
8	new	39.901945
9	reply	39.687749
10	ride	39.408613
11	good	38.269571
12	bikes	36.091646
13	think	35.315823
14	riding	33.196231
15	well	32.578283
16	time	32.033011
17	world	30.886983
18	usa	30.527954
19	much	30.124241

	term	tfidf (sum)
0	characteristics	0.034993
1	eventual	0.034993
2	approved	0.034993
3	pistons	0.034993
4	distances	0.034993
5	boiling	0.034993
6	unlike	0.034993
7	material	0.034768
8	ignoring	0.034768
9	imply	0.034768
10	trained	0.034768
11	corresponding	0.034768
12	empty	0.034768
13	regardless	0.034586
14	requires	0.034433
15	occasional	0.034433
16	preferable	0.034433
17	usual	0.034433
18	training	0.033759
19	cycle	0.033694

	term	tfidf (sum)
0	oriented	0.029919
1	civil	0.029919
2	standing	0.029764
3	signed	0.029764
4	rapidly	0.029764
5	regulations	0.029764
6	regarding	0.029764
7	williams	0.029634
8	crawl	0.029634
9	94	0.029634
10	mini	0.029522
11	fan	0.029522
12	followups	0.029522
13	utility	0.029522
14	seats	0.029336
15	popular	0.029336
16	defense	0.029257
17	thunder	0.028857
18	bird	0.028735
19	network	0.028431

Figure 3: tf-idf Sums of Weights, left-to-right: auto, cycle, auto, cycle

Extra Credit 1.

For an updated learning rate approach, we took the traditional "$\frac{1}{i}$" decay method. We defined our function as

$$\eta_i = \frac{\eta_0}{1.0 + \left(\frac{100 \cdot i}{\alpha \cdot 1192}\right)},$$

where α was the number of passes through the data, and i is the current iteration. The number '1192' is the total number of training examples (both train and test). The result of using a learning rate was a more steady convergence (even more so when we used tf-idf). A comparison can be found in Figure 4, with $\eta_0 = .1$, $\lambda = .00001$, and passes $= 10$. The corresponding labels are: ulr - updated learning rate, y - yes, and n - no.

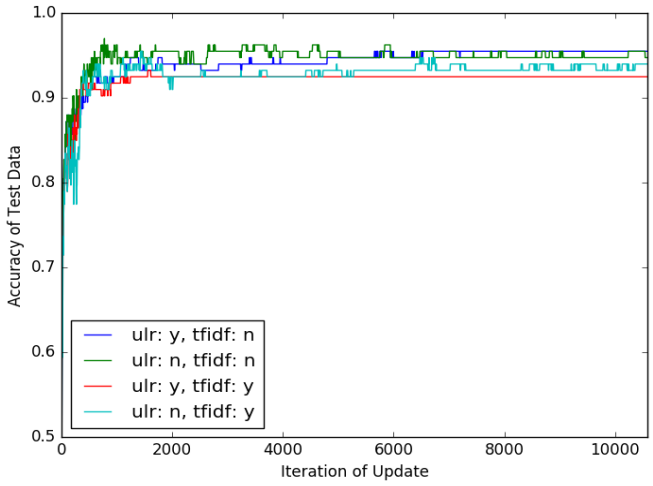

Figure 4: Scheduled Learning Rate Comparison

Extra Credit 2.

We will give two comparisons: 1) How tf-idf resulted in a more consistent convergence. 2) How the weighted sum of tf-idf compared with only tf in determining the best predictors for each class. For the first, we refer to Figure 5. There is less variance, and better convergence, over the majority of the tf-idf solutions, in comparison with the solutions in Figure 1. We now compare the best predictor words using tf-idf and only tf. See Figure 6 and 7 for the results. For auto, we see that there are some common words that ranked similar to both the tf-idf weighting and the flat tf count (car, like, cars). However, there are some terms in the flat tf count that did not appear in the tf-idf weight (go, people) and vice versa (ca, want). In the cycle count we have similar word rankings between tf-idf and tf (dod, bike, ride), others that appeared in the tf count and not the tf-idf weight (go, back), and vice versa (usa, world, bikes).

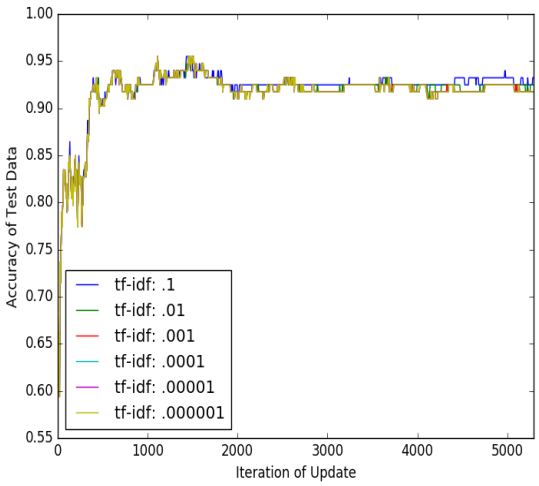

Figure 5: tf-idf

	term	tf
0	car	29500
1	like	19200
2	one	19100
3	distribution	17300
4	cars	16000
5	get	15400
6	good	14900
7	usa	14500
8	know	14200
9	think	13700
10	time	12600
11	also	12600
12	much	12500
13	new	12300
14	reply	11700
15	right	10400
16	go	10300
17	people	10300
18	please	10200
19	really	10100

Figure 6: auto term frequency

	term	tf
0	dod	27800
1	bike	23300
2	one	20800
3	like	20000
4	get	16500
5	know	14100
6	distribution	12900
7	ca	12900
8	good	11900
9	ride	11700
10	think	11200
11	new	10900
12	well	10800
13	time	10800
14	much	9800
15	go	9600
16	riding	9500
17	back	9400
18	reply	9400
19	right	9200

Figure 7: cycle term frequency