Final Report-Topic Modelling using Primal-Dual and Gradient-based algorithms for Non-negative Matrix Factorization

Ayush Jain, Akhil Israni

April 29, 2017

1 Introduction

In this project, we perform topic modeling of text data (short reviews) using Non-negative matrix factorization. NMF is a technique where a matrix V is factorized into (usually) two matrices W and H, with the property that all three matrices have no negative elements.NMF can be formulated as a minimization problem as shown below.NMF finds two non-negative matrices $W \epsilon R^{n \times r}$ and H $\epsilon R^{r \times m}$ such that

$$V \approx WH$$

Data Preprocessing:
1) A vector space model for each review is constructed (after removal of stopwords), resulting in a term-document matrix V from the complete set of reviews. In this step, rows correspond to the constructed vocabulary and columns includes the count of each term corresponding to each review.
2. TF-IDF [Term frequency–inverse document frequency] weight normalisation is applied to V. It is a numerical statistic that shows how important a word is to a review in a collection of reviews .The tf-idf value increases proportionally to the number of times a word appears in the document and how unique a term is across a set of reviews.
3. TF-IDF vectors in the matrix are normalized to unit length.
4. Then, using NMF [Non-negative matrix factorization], V matrix is factored into a term-topic (W) and a topic-document matrix(H).

Algorithms for NMF
5. We use a primal-dual algorithm [Ref.1] based on dual of KL-divergence cost function for NMF as well as certain gradient based approaches based on least square function (Frobenious norm) of NMF
6. Gradient based approaches includes gradient descent method with Armijo Rule and Lin Rule [Section 2.2 in Ref.2]

2 Primal-Dual Formulation

We use KL- Divergence as the cost function for NMF as it transforms into a convex decomposition problem.

$$D(V||WH) = -\sum_{i=1}^{m}\sum_{j=1}^{n} V_{ij}\left\{log\left(\frac{(WH)_{ij}}{V_{ij}}\right) + 1\right\} + \sum_{i=1}^{m}\sum_{j=1}^{n}(WH)_{ij} \tag{1}$$

$$\underset{W,H \geq 0}{\text{minimize}} \quad D(V||WH) \tag{2}$$

Problem is non-convex in both W and H simultaneously but convex in each factor separately:

$$\underset{W \geq 0}{\text{minimize}} \quad D(V||WH) \tag{3}$$

$$\underset{H \geq 0}{\text{minimize}} \quad D(V||WH) \tag{4}$$

A vector a ϵ R_+^p and a matrix $K\epsilon$ R_+^{pxq} as known parameters are considered, and x ϵ R_+^q as an unknown vector to be estimated, where the following expression holds,

$$a \approx Kx \tag{5}$$

and the aim is to minimize the KL divergence between a and Kx.
This is equivalent to a non-negative decomposition(ND) problem as defined in equations (3 and 4), considering a as a column of the given data, K as the fixed factor, and x as a column of the estimated factor.
In (3) equation, a and x are column vectors of V^T and W^T with the same index and K is $H^\top$, and in (4) equation, a and x are columns of V and H with the same index and K is W.

The convex Non-Decomposition problem with KL divergence is thus:

$$\underset{x \epsilon R_+^q}{\text{minimize}} \quad -\sum_{i=1}^{p} a_i(log(K_i x/a_i) + 1) + \sum_{i=1}^{p} K_i x \tag{6}$$

which may be written as :

$$\underset{x \epsilon \chi}{\text{minimize}} \quad F(Kx) + G(x) \tag{7}$$

By Fenchel's Duality Theorem [Theorem 3.10 in Ref.3] , the dual problem can be written as:

$$\underset{K^T(-y) \preceq K^T 1}{\text{maximize}} \quad a^T log(-y) \tag{8}$$

3 Optimization Methods

3.1 Primal-Dual algorithm

We apply the primal dual algorithm given in the Ref.1 for the same. This involves automatic heuristic selection of sigma and tau that are based on the primal-dual pairs. The pseudo code for the algorithm is shown in Figure 1.

3.2 Gradient-based approaches

We use Frobenious cost function of NMF [non-convex] directly for gradient based approaches.

$$\begin{aligned} \min_{W,H} \quad & f(W,H) = 1/2 \sum_{i=1}^{n} \sum_{j=1}^{m} (V_{ij} - (WH)_{ij})^2 \\ \text{subject to} \quad & W_{ia} \geq 0, H_{bj} \geq 0, \forall i, a, b, j \end{aligned} \tag{9}$$

This is Frobenius norm as shown below :

$$\sum_{i=1}^{n} \sum_{j=1}^{m} (V_{ij} - (WH)_{ij})^2 = ||V - WH||_F^2 \tag{10}$$

This technique is implemented for NMF based on the alternating procedure i.e., fixing one matrix, finding the other. The algorithm's pseudo-code is shown below where P(Δ) = max(0,Δ) is a projection function.

We are using function $F(W) = 1/2||V^T - H^T W^T||_F^2$
1. Initialize H^1 $\epsilon R_+^{MXR}, W^1 \epsilon R_+^{RXN}, 0 < \beta < 1, 0 < \sigma < 1$.

2.Iterations for q=1,2,....
(a) H^{q+1}= max(H^q-$\alpha_q \nabla_H F(H^q, W^q)$,0)

Select $\mathbf{V} \in \mathbb{R}_+^{n\times m}$, $\mathbf{W}_0 \in \mathbb{R}_+^{n\times r}$, and $\mathbf{H}_0 \in \mathbb{R}_+^{r\times m}$;

Set $\mathbf{W} = \bar{\mathbf{W}} = \mathbf{W}_{old} = \mathbf{W}_0$, $\mathbf{H} = \bar{\mathbf{H}} = \mathbf{H}_{old} = \mathbf{H}_0$, and $\chi = \mathbf{WH}$;

while *stopping criteria not reached* **do**

  Normalize $\mathbf{W}$ and set $\sigma = \sqrt{\frac{m}{r}}\frac{\mathbf{1}^\top\mathbf{H}\mathbf{1}}{\mathbf{1}^\top\mathbf{V}^\top\|\mathbf{H}\|}\mathbf{1}$, $\tau = \sqrt{\frac{r}{m}}\frac{\mathbf{1}^\top\mathbf{V}^\top}{\mathbf{1}^\top\mathbf{H}\mathbf{1}\|\mathbf{H}\|}\mathbf{1}$, and $\mathbf{H}(-\chi^\top) \le \mathbf{H}\mathbf{1}$;

  for $iter_{ND}$ *iterations* **do**

    $\chi^\top \leftarrow \chi^\top - \sigma \circ \left(\bar{\mathbf{W}}\mathbf{H}\right)^\top$;
    $\chi^\top \leftarrow \frac{1}{2}\left(\chi^\top - \sqrt{\chi^\top \circ \chi^\top + 4\sigma \circ \mathbf{V}^\top}\right)$;
    $\mathbf{W}^\top \leftarrow \left(\mathbf{W}^\top - \tau \circ \left(\mathbf{H}\left(\chi^\top + \mathbf{1}\right)\right)\right)_+$;
    $\bar{\mathbf{W}}^\top \leftarrow 2\mathbf{W}^\top - \mathbf{W}_{old}^\top$;
    $\mathbf{W}_{old}^\top \leftarrow \mathbf{W}^\top$;

  end

  Normalize $\mathbf{H}$ and set $\sigma = \sqrt{\frac{n}{r}}\frac{\mathbf{1}^\top\mathbf{W}\mathbf{1}}{\mathbf{1}^\top\mathbf{V}\|\mathbf{W}\|}\mathbf{1}$, $\tau = \sqrt{\frac{r}{n}}\frac{\mathbf{1}^\top\mathbf{V}}{\mathbf{1}^\top\mathbf{W}\mathbf{1}\|\mathbf{W}\|}\mathbf{1}$, and $\mathbf{W}^\top(-\chi) \le \mathbf{W}^\top\mathbf{1}$;

  for $iter_{ND}$ *iterations* **do**

    $\chi \leftarrow \chi - \sigma \circ \left(\mathbf{W}\bar{\mathbf{H}}\right)$;
    $\chi \leftarrow \frac{1}{2}\left(\chi - \sqrt{\chi \circ \chi + 4\sigma \circ \mathbf{V}}\right)$;
    $\mathbf{H} \leftarrow \left(\mathbf{H} - \tau \circ \left(\mathbf{W}^\top\left(\chi + \mathbf{1}\right)\right)\right)_+$;
    $\bar{\mathbf{H}} \leftarrow 2\mathbf{H} - \mathbf{H}_{old}$;
    $\mathbf{H}_{old} \leftarrow \mathbf{H}$;

  end

end

return $\mathbf{W}^\star = \mathbf{W}$, *and* $\mathbf{H}^\star = \mathbf{H}$.

(a) Primal-Dual algorithm

Figure 1: Primal-Dual algorithm

where $\alpha_q = \beta^{t_q}$, and t_q is the first non negative integer for which the below condition is satisfied.

(b) W^{q+1}= max(W^q-$\alpha_q \nabla_Q F(H^{q+1}, W^q)$,0)
where $\alpha_q = \beta^{t_q}$, and t_q is the first non negative integer for which the below condition is satisfied.

The Armijo rule uses the condition:

$$(1-\sigma)\ \langle \nabla_H F(W^q, H^q), H^{q+1} - H^q\rangle + 1/2\langle H^{q+1} - H^q, ({W^q}^T W^q)(H^{q+1} - H^q)\rangle \le 0$$

This condition ensures that there is a sufficient decrease at each iteration providing a good convergence performance.

Lin Method- In this method, we use $\alpha_{(q-1)}$ as an initial guess for α_q, which has the advantage of taking fewer steps to find α_q. If α_q satisfies the above condition, repeatedly do $\alpha_q = \alpha_q/\beta$ until it does not satisfy the above condition or H_q remains unchanged when updating α_q Else repeatedly decrease α_q by $\alpha_q = \alpha_q \cdot \beta$ until α_q satisfies the above condition.

4 Experimental Results

In order to ensure the correctness of implemented algorithms, we tested the algorithms for synthetic data which involved a Matrix V of size 500x500 randomly generated from the uniform distribution. The low-rank element(r) which corresponds to the number of topics, was set to r = 10 and r=30. W_0 and H_0 are initialized element-wise using a random number in range (0,1).

The graphs for the experiments are shown below. We used L_2 norm of matrix (V-WH)as a measure of how close we reach the optimum value.
We conducted four experiments for the three algorithms:
i) **Objective function vs No of iterations**
In this experiment as shown in Figure 2 and 3, we vary the number of iterations from 10 to 2000 for the three algorithms to see the trend in the value of L_2 norm. As expected, it decreases showing the convergence with the increase in number of iterations. We are gettting a sharp knee at n=100 iterations for all three algorithms for the random data. Whereas for the Amazon reviews data set we see that Lin algorithm converges much faster than the other two algorithms with knee point a close to n=25 iterations.Refer figure 2,3
ii) **Size of Dataset vs Algorithm effectiveness**
For dataset of size 100x100, the vocabulary consists of very few features which makes it difficult to interpret the topic based on the words associated with it. Moreover, the topic clusters of document are vague as well. Thus, even if the optimization algorithm are accurate the sparse representation results in poor modelling. For dataset of size 500x500, the vocabulary size is decent enough to interpret the results of matrix factorization as can be seen in inference section.
iii) **Comparison of dual approach and gradient based algorithms**

Algorithm	Optimum	number of iterations for Knee Point	Rank of Matrix
Armijo	27.68	100	10
Lin	15.34	50	10
Primal Dual	19.88	100	30

Table 1: Comparison of Three Algorithms for random data

Algorithm	Optimum	number of iterations for Knee Point	Rank of Matrix
Armijo	1.9	25	10
Lin	1.85	25	10
Primal Dual	0.73	500	50

Table 2: Comparison of Three Algorithms for Amazon data

As we can see in the tables the performance of both the gradient based algorithms(armijo and lin) are almost similar for both the random as well as amazon reviews dataset. Also for the amazon reviews dataset we can see that the optimum value achieved by the primal dual algorithm is better than the gradient based algorithms but it takes more number of iteration to converge and hence is slower than gradient based algorithms.
The gradient based algorithms are well suited for low rank matrices where rank is 10or 30, whereas the primal dual algorithm performs better for higher rank matrices near to 50.
iv)**Objective function vs Rank of matrices**.
In this experiment as shown in Figure 4, we varied the choice of rank of matrices W and H from 10 to 200. The optimum value achieved for both the algorithms was better(lower) when we use a low rank matrix like [10,30] than when we used a higher rank matrix [100,200].
v)**Inference from Amazon Reviews Dataset**
As we can see from the above experiments, all the algorithms achieve optimum values close to Rank - 30 and rank - 10 for matrix size 500x500 . Here row denotes a document as a tf-idf vector and column represents the vocabulary for the 500 documents after removal of stop words. Based on Non-negative Matrix Factorization of V, we get W and H where W denotes term-topic matrix and H denotes document-topic matrix(describing data clusters of related documents referring to a particular topic).

We interpret the results of V and H obtained with Primal-Dual algorithm (as it gives the lowest norm value) to see if the words associated with topic makes sense semantically and documents with similar ideas map to particular idea. Since the optimum values after applying gradient based approach is close to Primal-Dual algorithm, the interpretation can be extended to their NMF results as well.
For Matrix of size 500x500, the vocabulary size is 500 and consists of words like
{'quickly','quicktime','quite','quot','ram','read','reading','real' 'really','reason','received','recently',
'recommend','review' 'reviews','right','rom','room','run','running','said','save' 'say','saying','says',
'school','screen','second','section','seen' 'send','set','short','shows','sign','sim',
'simple','simply' 'single','sister','site','skating','skills','slide','slow','small' 'smartsuite',
'software','son','song','sound','space','speed' 'spend','spent','spreadsheet','standard',
'star','stars','start' 'started','student','study','stuff','suite','support','sure' 'takes','talk',
'tasks','teach','teaches','teaching','tech', 'technical'}

For rank =10, the topics and their associated words are shown below based on inference from matrix W:
Topic 0:
office word mac microsoft os entourage excel powerpoint version windows features ms use files apple suite work open need versions
Topic 1:
game fun games play kids like cool love think really played playing buy easy loved graphics hard just child son
Topic 2:
language learn sign words learning french phrases dictionary japanese disk fun immersion grammar instant people basic set learned vocabulary class
Topic 3:
biology biotutor grade bio tutor excalibur student helped class help program software extra school doing questions great notes high tests
Topic 4:
program software just use time computer like work support works good don xp new ve better tried using does did
Topic 5:
old year loves game daughter worth love parade play likes girl decorating castle great maze barbie girls playing song played
Topic 6:
travel guide garmin information money major limited waste expensive card data 50 lot number useful worth business gives save don
Topic 7:
product excellent easy great use got worth good quot money version information tell like used features bit language amazon does
Topic 8:
cd rom received instead words cds french worked recommend italian quot teaches exactly drive time easy second order son including
Topic 9:
barbie music ice choose pick skating shows tour different favorite daughter color christmas wonderful beautiful levels room create highly child

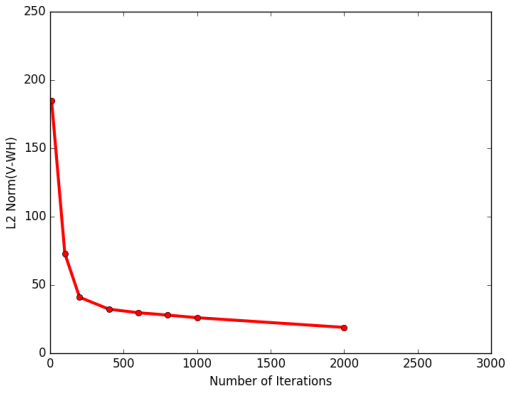
We can also identify topic associated document based on inference from matrix H. For topic 0, a sample associated document is as follows:
"Microsoft Office v.X gets a BUY recommendation. If thats all you want to know, then get
online and buy this beast!I call it a beast affectionately, but lets face it. If you didn't need to bring work home from the office, would you REALLY be buying a program with a word processor, spreadsheet, slide show and mail program? Probably not. However, having said that, if you were to consider buying any ONE of these programs separately, the cost difference between one and all four is such that any frugal buyer would opt for Office just on that basis."
As it can be seen, the review talks about microsoft office products and it's features which highly relates to **Topic 0**.
Similarly, for topic 1, a sample associated document based on inference from matrix H is as follows:

"My three year old adores this game. The game only takes 2-3 hours to complete in one sitting, but my child has played it over and over, and still loves it. The box recommends this game for ages 5 and older, but my daughter can play this game by herself with ease, except for the maze (which is fairly easy, but she needs help working the arrow keys). This game is mostly a "paint" type game, where you decorate different rooms in a castle in order to save Prince Stephan and the castle. You paint and place different decorations about the castle, as well as "build" a planter and a mosaic tile. The game and the decorations are quite cute, and easy for young Barbie fans to do independently. I feel it does have some educational value as well, in that my daughter seems to have improved her mouse skills from playing this game, has grasped concepts like clicking a "done" button when she has completed a task, and seems to have become better at listening to directions."
Here, the review talks about playing games for a 3-year old which highly relates to **topic 1.**

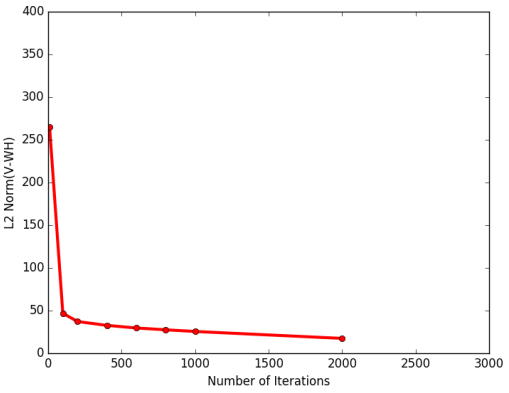
5 Conclusion

Thus, it can be seen that the matrix factorization are giving us the correct results and by minimizing the L-2 norm using our gradient based approaches and primal-dual method, we optimize the accuracy of matrix W and H giving us semantically correct results. Out of the algorithms , the gradient based approach with lin method converges the fastest but the minimum optimum value is given by primal-dual showing its effectiveness over others in terms of accuracy if not speed.

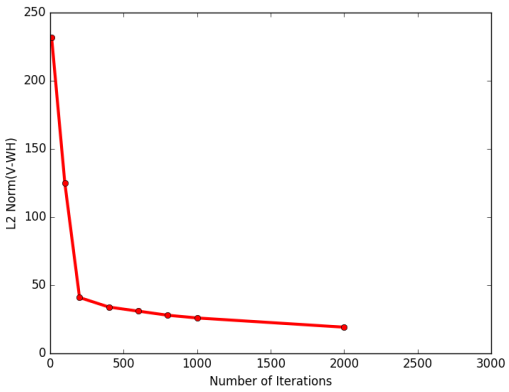
After preprocessing of our own dataset, We ran the same experiments for the three algorithms on Amazon software reviews dataset as well. We used around 500 dimensional data(around 500 terms in the vocabulary) and 500 reviews for preliminary experiments and the trend of L2 norm with the increase in the number of iterations can be seen in Figure 5. As seen, both gradient descent algorithm start to converge at a much faster rate than Primal-dual algorithm.



(a) Armijo Algorithm Rank 30 Random Data

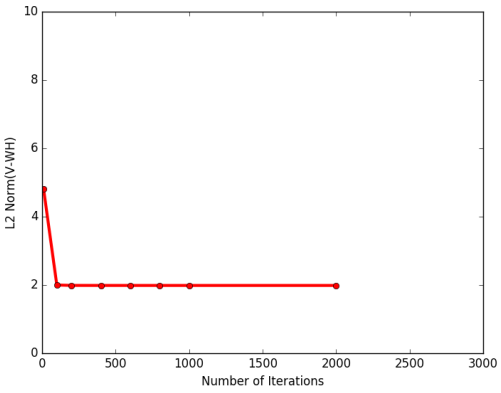


(b) Lin Algorithm Rank 30 Random Data

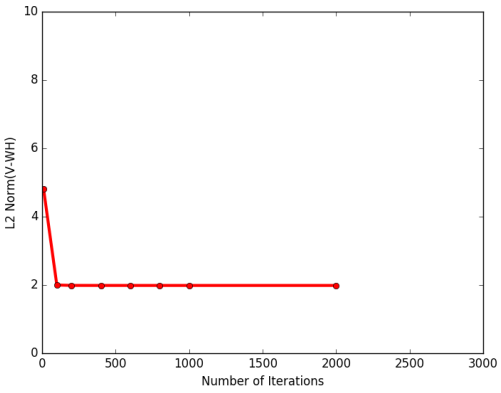


(c) Primal dual Algorithm Rank 30 Random Data

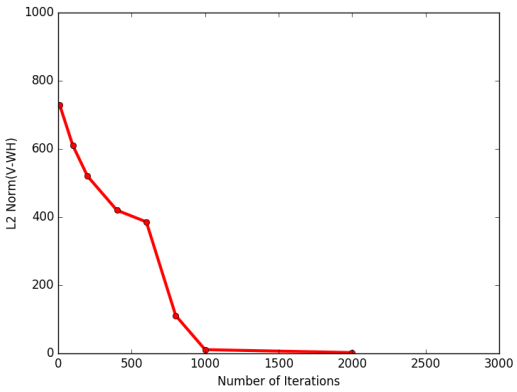
Figure 2: Performance of All algorithms on Random Dataset



(a) Armijo Algorithm Rank 30 Amazon Data

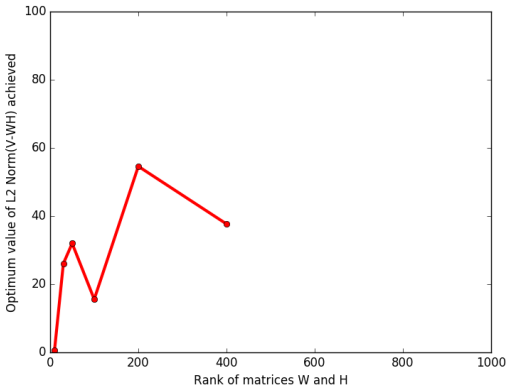


(b) Lin Algorithm Rank 30 Amazon Data

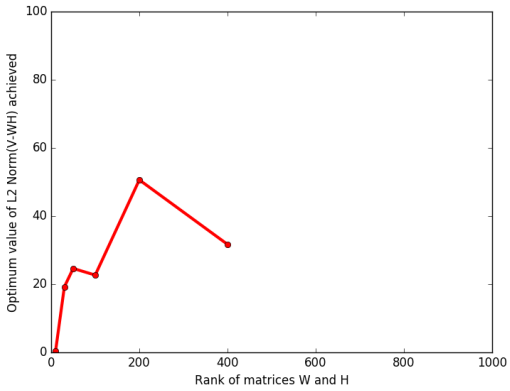


(c) Primaldual Algorithm Rank 30 Amazon Data

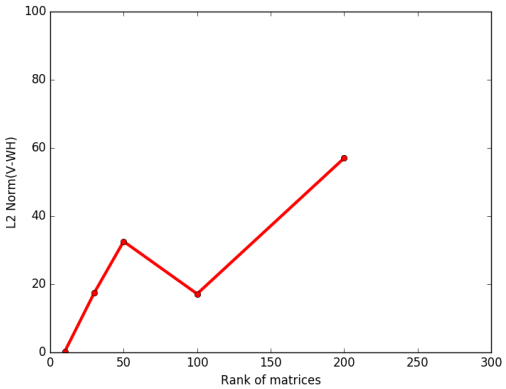
Figure 3: Performance of All algorithms on Amazon Dataset



(a) Armijo Algorithm- rank vs Optimum Value achieved

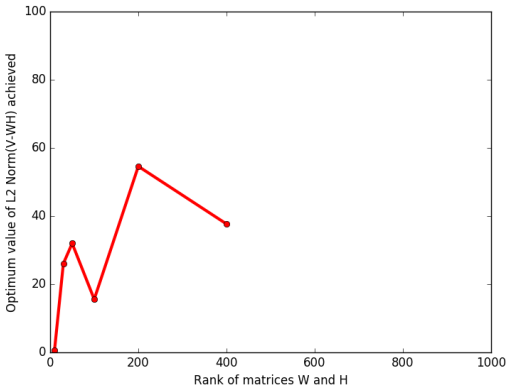


(b) Primal-Dual algorithm rank vs Optimum Value achieved

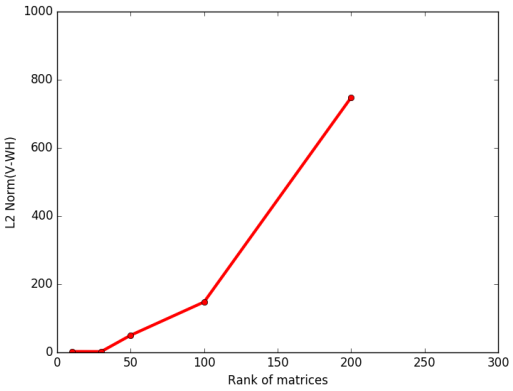


(c) Lin algorithm rank vs Optimum Value achieved

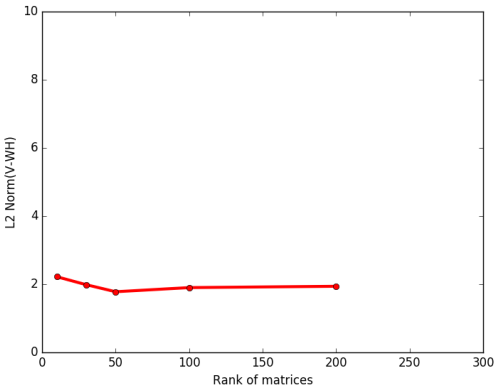
Figure 4: Rank vs Optimum Value achieved(Random Data set)



(a) Armijo Algorithm- rank vs Optimum Value achieved



(b) Primal-Dual algorithm rank vs Optimum Value achieved



(c) Lin algorithm rank vs Optimum Value achieved

Figure 5: Rank vs Optimum Value achieved(Amazon Data set)

6 Dataset Description and Implementation Details

The dataset (Software.tar.gz) consists of around 95k reviews on Software products and they are converted into the required matrix V as per our methodology using our preprocessing script script.py [in python]. The algorithms have been implemented in MATLAB as pgwitharmijo.m ,dualkl.m and pgwithlin.m respectively. The matrix V from preprocessing is imported in driver.m for various experiments.
Dataset - Amazon Reviews on Software : https://snap.stanford.edu/data/web-Amazon.html
Reference 1 - : (http://matlabtools.com/wp-content/uploads/pro409.pdf)
Reference 2 - : (http://info.ee.surrey.ac.uk/CVSSP/Publications/papers/WangZ-ICARN-2008.pdf)
Reference 3 - : (http://num.math.uni-goettingen.de/~r.luke/publications/Handbook_BorweinLuke.pdf)