

Math 320 Project

Anh Phan

Nonnegative Matrix Factorization and SVD - how Linear Algebra get into Topic Modeling

1 Introduction

Natural Language Processing models have always been getting a lot of interest as the ability to work with text as data and apply different mathematical methods has always been useful in a variety of tasks, such as document classification, document clustering, text generation, question answering, etc. Among these tasks, one task that has been focused and gained interest steadily over time is the area of document clustering, or topic modeling, where your task is to try to group a series of documents into a few "topics", where from here, you can explore different structure behind these documents. Imagine you have a series of notes (a lot of notes) from different classes, meetings, activities, etc. that you have forgotten for a long time and you want to understand what type these notes are in. Since you might have forgotten what major topics are these notes from, topic models could be a solution to this, without you having to do manual work. That is the reason why there have been many topic models that have been developed over the years: from simpler ones like the Latent Dirichlet Allocation (2003) and Structural Topic Model (2014) to more complex and modern ones like the Embedded Topic Model(2020), BERTopic (2022).

Among these models, one simpler method that has been working well across different scenarios is based on a unique method of matrix decomposition/approximation, which is the Nonnegative Matrix Factorization (NMF), which has been applied in topic modeling for a long time (Xu et al., 2003). In this paper, we will discuss this unique method and the unique connection of this method to a widely-used matrix factorization, the Singular Value Decomposition (SVD).

2 Nonnegative Matrix Factorization

NMF was discovered in 1999 with the idea of decomposing matrix data to work with a lower-rank matrix, and exploring the structure behind data. The problem associated with this algorithm is as follows:

Given a matrix A of size $m \times n$ having nonnegative entries and a number k , find matrices W, H with nonnegative entries of size $m \times k$ and $k \times n$ such that:

$$\|A - WH\|_F = \sum_{i=1}^m \sum_{j=1}^n (a_{i,j} - W_{i,*} H_{*,j})^2 \quad (1)$$

is minimized toward 0 ($M_{i,*}$ is the i^{th} row of M and $M_{*,i}$ for the i^{th} column).

In layman's terms, this problem is equivalent to finding the closest "decomposition" of a nonnegative matrix A into two other nonnegative matrices of defined size. Here, the value of m, n will often be very high and the k value will often be lower, so that W, H will have a simpler structure with a lower rank, which allows us to analyze more efficiently, and this method will also work well in cases where the context of data we analyze is mostly nonnegative (like in chemometrics or processing audio spectrograms).

From the structure of the output, researchers also discovered the clustering property behind it, and in a special case when we assume H to have the orthogonal property ($HH^T = I$), then it is equivalent to some famous clustering method like K-means clustering (Ding et al., 2015). The entries of H could also suggest what data point is in what cluster, for example, in i^{th} column of matrix H , the dominant entries suggest what cluster the i^{th} data point is in (if the j^{th} entries in column i is the largest entries in column i , then the i^{th} data point is likely to be in the j^{th}).

While this seems to be a very useful method, its strict condition on the positivity of the resulting matrices also poses some major theoretical issues. One of the major issues is finding the "exact" solution to this problem, which means finding nonnegative matrices W, H such that $A = WH$, is an NP-hard problem, which is an extremely hard problem that may be even undecidable (Vavasis, 2008.) Another issue is that the approximation might not be unique (Xu et al., 2003.) Due to this, most methods for finding W, H are often heuristic methods that may find W, H that satisfy:

$$\|A - WH\|_F < \epsilon \quad (2)$$

for some small ϵ .

In order to optimize the function in (1), most methods will follow a framework of initializing some nonnegative matrices W_0, H_0 , then for each iteration, we will update to make W_{new}, H_{new} based on W_{old}, H_{old}, A such that the function in (1) will decrease (Gillis, 2014). The algorithm will run until a certain condition has been reached. One of the most famous method that follow this framework is the multiplicative update method proposed by Lee and Seung in 2001:

Algorithm 1: NMF estimation based on Lee and Seung Multiplicative Update Rule

Input: Nonnegative matrix A , positive integer k, t , positive real number ϵ

Output: Nonnegative matrices W, H

1 Initialize W_0, H_0 nonnegative

2 For $i = 1, 2, \dots, t$:

- Calculate W_{i+1}, H_{i+1} based on these rules (Note that \circ is element-wise matrix multiplication and \oslash is element-wise division of 2 matrices):

$$H_i = H_{i-1} \circ \frac{[W_{i-1}^T A]}{[W_{i-1}^T W_{i-1} H_{i-1}]} \quad (3)$$

$$W_i = W_{i-1} \circ \frac{[A H_i^T]}{[W_{i-1} H_{i-1} H_i^T]} \quad (4)$$

- Check if this inequality satisfies:

$$\|A - W_i H_i\|_F < \epsilon \quad (5)$$

if yes, then we stop the algorithm, otherwise, we continue the algorithm.

As explained in a paper by Nicholas Gillis in 2014, the authors of this multiplicative rule developed this rule based on the optimality conditions of the objective function. Let

$$f(W, H) = \|A - WH\|_F \quad (6)$$

be our objective function that we need to minimize, where $W, H \geq 0$, then we can use the Karush-Kuhn-Tucker condition (Karush et al., 1939), the optimal conditions are:

$$\nabla_W f(W, H) = 2WHH^T - 2AH^T \geq 0 \quad (7)$$

$$\nabla_H f(W, H) = 2W^T WH - 2W^T A \geq 0 \quad (8)$$

$$W \circ \nabla_W f(W, H) = 0 \quad (9)$$

$$H \circ \nabla_H f(W, H) = 0 \quad (10)$$

where $\nabla_X Y$ is the gradient based on a variable matrix (for each element at position (i, j) , we take $\frac{dy_{i,j}}{dx_{i,j}}$

In addition to Lee and Seung’s multiplicative update rule, some other methods were also discovered to optimize the NMF objective function in a better manner. One method that is worth mentioning due to its simplicity and its close relation to one of the most basic problems in mathematics optimization, the Linear Least Square algorithm, is the Alternating Least Square method, which was first mentioned in a paper by Berry et al. in 2007. Here, the main idea is that we keep doing linear least square and eliminate negative entries, the algorithm follows these steps (Berry et al., 2007):

Algorithm 2: Alternating Least Square

Input: Nonnegative matrix A , positive integer k , t , positive real number ϵ

Output: Nonnegative matrices W , H

1 Initialize a nonnegative matrix W

2 For $i = 1, 2, \dots, t$:

- Solve the least square equation for H

$$W^T W H = W^T A \quad (11)$$

- Remove all negative entries in H

- Solve the least square equation for new W

$$H^T H W = H^T A \quad (12)$$

- Check if this inequality satisfies:

$$\|A - W_i H_i\|_F < \epsilon \quad (13)$$

if yes, then we stop the algorithm, otherwise, we continue the algorithm.

While these are some decent methods for estimating the matrices for NMF, one issue still remains, how do we initialize W_0, H_0 , as initializing randomly often hard to converge and the solution is not reproducible (Hafshejani et al., 2021). This raises another problem of initializing NMF more systematically, and among various methods that have been developed, one method that combines an advanced linear algebra method stood out, that is Nonnegative Double Singular Value Decomposition (NNDSVD) (Boutsidis et al., 2008). This method utilizes a more familiar SVD decomposition and its expansion, the truncated SVD, as a way to initialize W_0, H_0 with a certain rank, in which their product is close to A . In the next part, we will dive into how SVD and NMF come together to make a more complete algorithm.

3 NNDSVD: how SVD makes NMF more stable

Before we go into how the SVD method is used in the initialization of the NMF algorithm, we first need to look back at the SVD itself. One basic representation of SVD is as follows: Given a matrix A of rank r , we can represent A as follows:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (14)$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ are r non-zero singular values of A , and for each σ_j , u_j, v_j are the left and right singular vectors associated with it. From here, the best rank $q \leq r$ approximation to A is:

$$A_q = \sum_{i=1}^q \sigma_i u_i v_i^T \quad (15)$$

If we let $M_i = u_i v_i^T$ for $i = 1, 2, \dots$, then we can write

$$A = \sum_{i=1}^r \sigma_i M_i \quad (16)$$

Another idea that we also consider is that given a matrix M , we define the positive part and negative part of M to be nonnegative matrices M^+ and M^- (we also define positive and negative of vector in a similar manner) ($M = M^+ - M^-$), where:

$$M_{x,y}^+ = \begin{cases} M_{x,y}, & \text{if } M_{x,y} \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

$$M_{x,y}^- = \begin{cases} |M_{x,y}|, & \text{if } M_{x,y} < 0 \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

From here, if we let $B \geq 0$ and consider $f(B) = \sum_{i=1}^m \sum_{j=1}^n (a_{i,j} - b_{i,j})^2$, then we can minimize $f(B)$ by minimizing each square, as if $a_{i,j} < 0 \leq b_{i,j}$ then $(a_{i,j} - b_{i,j})^2 \geq (a_{i,j} - 0)^2 = a_{i,j}^2$, and if $a_{i,j}, b_{i,j} \geq 0$ then $(a_{i,j} - b_{i,j})^2 \geq (a_{i,j} - a_{i,j})^2 = 0$. Therefore, we will have the best nonnegative approximation of M in terms of Frobenius norm is M^+ (this is also mentioned in Boutsidis et al. paper as a basis for the NNDSVD method).

Another important idea in this method is the nonnegative estimation of rank 1 matrices using SVD. A key idea is that for a matrix of rank 1, we can find its singular values for its positive and negative parts, which then allows us to find a rank 1 nonnegative approximation to this matrix. The theorem follows this claim (proved in the Boutsidis paper):

Given a matrix M of rank 1 (we can write it as $M = xy^T$), and define: $\hat{x}^+, \hat{x}^-, \hat{y}^+, \hat{y}^-$ be the normalized version of positive and negative parts of x, y , and let $\mu^+ = \|\hat{x}^+\| \|\hat{y}^+\|$, $\mu^- = \|\hat{x}^-\| \|\hat{y}^-\|$, $\xi^+ = \|\hat{x}^+\| \|\hat{y}^-\|$, $\xi^- = \|\hat{x}^-\| \|\hat{y}^+\|$. Then M^+ will have singular values of μ^+, μ^- and M^- will have singular values of ξ^+, ξ^- , and we can write:

$$M^+ = \mu^+ \hat{x}^+ (\hat{y}^+)^T + \mu^- \hat{x}^- (\hat{y}^-)^T \quad (19)$$

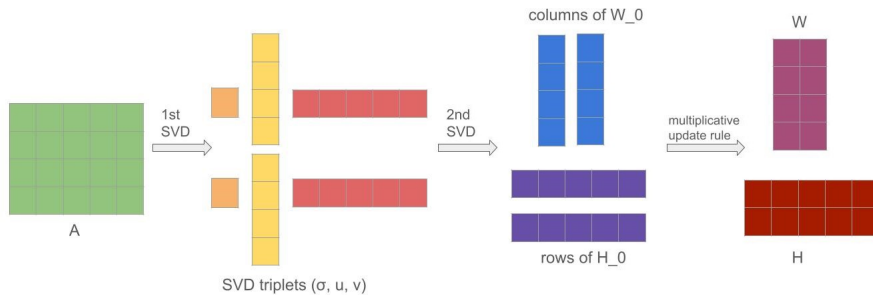
$$M^- = \xi^+ \hat{x}^+ (\hat{y}^-)^T + \xi^- \hat{x}^- (\hat{y}^+)^T \quad (20)$$

From this, we can go back to our truncated SVD equation (15), and notice that we can have rank 1 estimation to each M_i as follows (expressions defined as previous)

$$M_i \approx M_i^+ \approx \begin{cases} \mu_i^+ \hat{u}_i^+ (\hat{v}_i^+)^T, & \text{if } \mu_i^+ > \mu_i^- \\ \mu_i^- \hat{u}_i^- (\hat{v}_i^-)^T, & \text{otherwise} \end{cases} \quad (21)$$

From here we can apply this estimation to all components in the rank-k truncated SVD. And now we constructed the original NNDSVD algorithm as follow (here v^* is the conjugate transpose of v , and we also define all the expressions similar to previous ones):

Figure 1: NMF with NNDSVD initialization and multiplicative update rule



Algorithm 3: Nonnegative Double SVD algorithm

Input: Nonnegative matrix A , positive integer k **Output:** Rank k nonnegative matrices W, H 1 Find the rank k truncated SVD of A :

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T \quad (22)$$

2 Let $w_1 = \sqrt{\sigma_1} |u_1|$ and $h_1 = \sqrt{\sigma_1} |v_1^*|$ 3 For $i = 2, \dots, k$:- Let $x = u_i, y = v_i$ - If $\mu^+ > \mu^-$, then $u = \hat{x}^+, v = \hat{y}^+, \sigma = \mu^+$ - otherwise, $u = \hat{x}^-, v = \hat{y}^-, \sigma = \mu^-$ - Let $w_i = (\sqrt{\sigma_i \sigma}) u$ and $h_i = (\sqrt{\sigma_i \sigma}) v^*$

4 Set

$$W = [w_1 \quad \dots \quad w_k] \quad (23)$$

$$H = [h_1 \quad \dots \quad h_k]^T \quad (24)$$

The second SVD step on rank 1 matrices has gained the method name "double SVD". From here, we have constructed rank k matrices W, H that are both nonnegative for initialization of NMF, and since this algorithm is not randomized, we can be more confident on our result of further iterations of the NMF algorithm. Here we can see from Figure 1 how the method would look from start to end.

4 NMF and Topic Modeling

As we have developed a better understanding of NMF, it is important to discuss its application in topic modeling. To explore this application, we will have to look at one method to convert text to data: the bag of word method

In the bag of word approach, we will consider each unit of documents (can be a sentence, a paragraph, or even a document) as a collection of words, without considering their order. So we will have for each document its words and the count of each word in the document. From here, we can group all documents to create a term-document matrix:

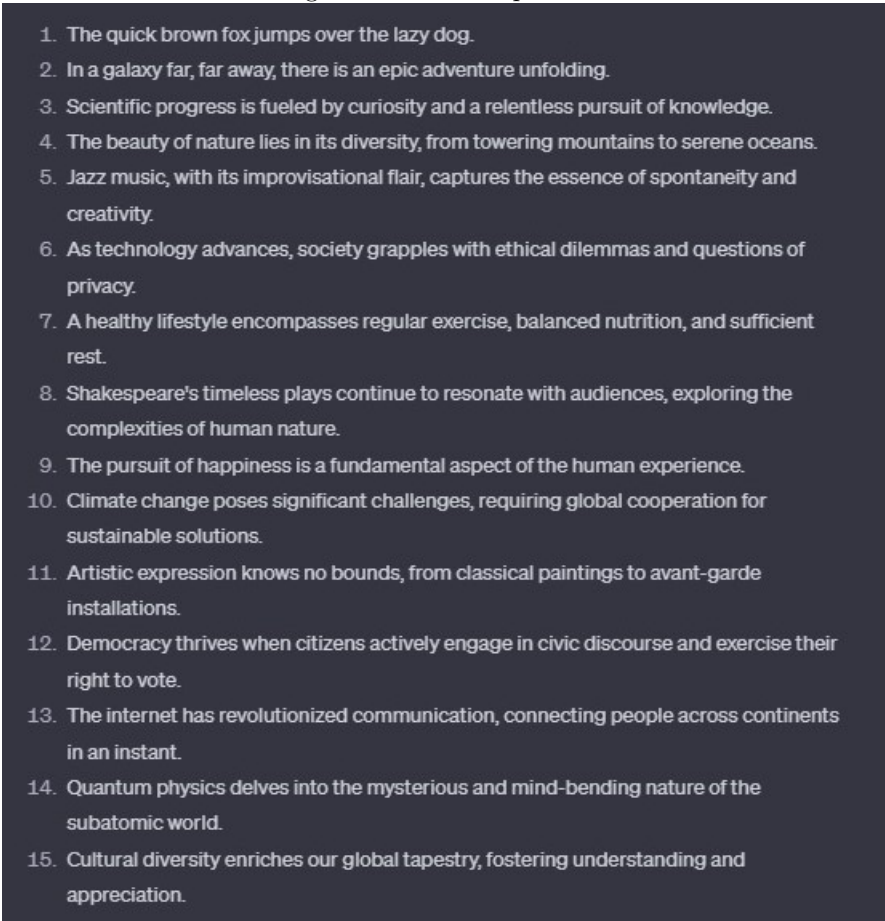
Figure 2: NMF with NNDSVD initialization and multiplicative update rule

Venue	CAMAD	EUNICE	HAISA	HPCC-ICESS	IESMA	ISCA	KMIS	NMR	SPRINGL	SSV
Keyword										
algorithm	2	8	0	24	0	5	0	2	1	1
cellular	2	1	0	1	0	0	0	0	0	0
game	1	1	0	1	0	0	1	1	0	0
hardwar	1	0	1	4	0	18	0	0	1	0
internet	2	6	2	0	2	0	0	0	0	0
mobil	10	8	0	6	17	5	2	0	2	0
network	58	60	4	38	2	25	12	0	3	0
search	0	1	0	1	2	4	1	0	0	0
secur	4	4	29	5	1	12	3	0	4	0
web	0	2	0	3	3	1	13	0	2	0

From here, we will convert our term-document matrix to represent the frequency of a word in a document, and we use the resulting matrix as the input matrix A in NMF, and the number of topics that we would want to cluster into, k , would be the other parameter. After we run the NMF algorithm, we will get two matrices: W , which will represent the word distribution of each of the k topics, while H will represent the proportion of each topic in each document (or topic prevalence of each document.) Because of the goal of describing the corpus (collection of documents) as topics' word distribution and topic prevalence, the need for nonnegative matrix factorization arises, and that explains why NMF is still being used until now for topic modeling.

In order to demonstrate further, we will have a simple example of implementing the NMF method with NNDSVD initialization on a collection of text (the source code will be at my Github). We first start by getting some generated data of text of different contexts:

Figure 3: Our example data

- 
1. The quick brown fox jumps over the lazy dog.
 2. In a galaxy far, far away, there is an epic adventure unfolding.
 3. Scientific progress is fueled by curiosity and a relentless pursuit of knowledge.
 4. The beauty of nature lies in its diversity, from towering mountains to serene oceans.
 5. Jazz music, with its improvisational flair, captures the essence of spontaneity and creativity.
 6. As technology advances, society grapples with ethical dilemmas and questions of privacy.
 7. A healthy lifestyle encompasses regular exercise, balanced nutrition, and sufficient rest.
 8. Shakespeare's timeless plays continue to resonate with audiences, exploring the complexities of human nature.
 9. The pursuit of happiness is a fundamental aspect of the human experience.
 10. Climate change poses significant challenges, requiring global cooperation for sustainable solutions.
 11. Artistic expression knows no bounds, from classical paintings to avant-garde installations.
 12. Democracy thrives when citizens actively engage in civic discourse and exercise their right to vote.
 13. The internet has revolutionized communication, connecting people across continents in an instant.
 14. Quantum physics delves into the mysterious and mind-bending nature of the subatomic world.
 15. Cultural diversity enriches our global tapestry, fostering understanding and appreciation.

From here, we will clean the dataset (removing punctuation, stopwords, etc.), as well as get the unique list of words that are in these chunks of text. Then we will create the frequency term-document matrix for the input of the NMF algorithm. Then we conduct the algorithm on a different number of topics, number of iterations, and epsilon values, using the scheme as discussed above. A few interesting findings that we found from running the models is when we consider 4 topics in our text. We have divided documents into each of the topics as follows, and in addition to this, we are also able to extract the top words in each of the topics, which might explain why the topics are divided as in Figure 4 (each collection of top words seems to represent a theme):

Figure 4: Division of documents into topics

```
documents in topic 1
scientific progress is fueled by curiosity and a relentless pursuit of knowledge
jazz music with its improvisational flair captures the essence of spontaneity and creativity
as technology advances society grapples with ethical dilemmas and questions of privacy
a healthy lifestyle encompasses regular exercise balanced nutrition and sufficient rest
the pursuit of happiness is a fundamental aspect of the human experience
artistic expression knows no bounds from classical paintings to avant-garde installations
democracy thrives when citizens actively engage in civic discourse and exercise their right to vote
the internet has revolutionized communication connecting people across continents in an instant

documents in topic 2
in a galaxy far far away there is an epic adventure unfolding

documents in topic 3
the quick brown fox jumps over the lazy dog

documents in topic 4
the beauty of nature lies in its diversity from towering mountains to serene oceans
shakespeare's timeless plays continue to resonate with audiences exploring the complexities of human nature
climate change poses significant challenges requiring global cooperation for sustainable solutions
quantum physics delves into the mysterious and mind-bending nature of the subatomic world
cultural diversity enriches our global tapestry fostering understanding and appreciation
```

Figure 5: Top words of each topic

```
top words in topic 1 with probability
['pursuit', 'human', 'fundamental', 'happiness', 'aspect']
top words in topic 2 with probability
['far', 'away', 'adventure', 'galaxy', 'epic']
top words in topic 3 with probability
['fox', 'jumps', 'quick', 'lazy', 'dog']
top words in topic 4 with probability
['nature', 'diversity', 'towering', 'serene', 'lies']
```

As mentioned, as we look at how the topics have some distinction in word choices of the meaning of the sentences, we can see that the NMF model works to some extent with a rather small collection. Our NMF factorization of the frequency term-document matrix also has a mean difference in entries with the original matrix of 0.00066, which is very small considering the scale of the matrix. By using more iterations, more data, and a better number of topics, we might achieve better results than our current one. From here, we have seen how NMF could be applied in a real situation in topic modelings, and this method can be scaled to apply to datasets of much larger size.

5 Conclusion

Nonnegative Matrix Factorization has proved to be a useful and easy-to-implement method for a variety of areas, especially in topic modeling, and the discovery of SVD-related initialization methods in an attempt to stabilize the method, with the pioneer one to be Nonnegative Double SVD, has helped the whole framework in general to be welcomed and used more and more by the community. While there are still some issues with this problem (hard to estimate, may have more than 1 solution, etc.), as the area of NMF is still a very interesting area, especially in terms of finding exact form for some types of matrices, there might be more and more algorithm for finding NMF in the future, with some prospect of a perfect decomposition algorithm like how SVD or eigendecomposition has been discovered. Overall, the NMF with SVD-based initialization is still a useful method and have a lot of prospect in the future.

Appendix

- Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., & Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1), 155–173. <https://doi.org/10.1016/j.csda.2006.11.006>
- Boutsidis, C., & Gallopoulos, E. (2008). SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4), 1350–1362. <https://doi.org/10.1016/j.patcog.2007.09.010>
- Ding, C., He, X., & Simon, H. D. (2005). On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. *Society for Industrial and Applied Mathematics*.
- Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1, 211–218. <https://doi.org/10.1007/BF02288367>
- Gillis, N. (2014). The Why and How of Nonnegative Matrix Factorization. *ArXiv.Org*. <https://doi.org/10.48550/arxiv.1401.5226>
- Hafshejani, S. F., & Moaberfard, Z. (2021). Initialization for Nonnegative Matrix Factorization: a Comprehensive Review. *ArXiv.Org*. <https://doi.org/10.48550/arxiv.2109.03874>
- Karush, W. (1939). Minima of functions of several variables with inequalities as side conditions.
- Seung, H. S., & Lee, D. D. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature (London)*, 401(6755), 788–791. <https://doi.org/10.1038/44565>
- Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13 - Proceedings of the 2000 Conference, NIPS 2000 (Advances in Neural Information Processing Systems)*. Neural information processing systems foundation.
- Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. *Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*; 28 July-01 Aug. 2003, 267–273. <https://doi.org/10.1145/860435.860485>