International Journal of Computing and Optimization
Vol. 7, 2020, no. 1, 47 – 56
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ijco.2020.9825

A Penalty Trust Region Method for Nonnegative Matrix Factorization

Chaofeng Ye, Tao Yan and Kai Wang

Department of Mathematics
College of Science
Nanjing University of Science and Technology
Nanjing, Jiangsu 210094, China

Abstract

Nonnegative matrix factorization (NMF) is to decompose a nonnegative matrix into the product of two smaller nonnegative matrices. It is one of the popular ways for dimension reduction in data processing. In this paper, we firstly reduce the dimension of the original matrix by using the orthogonal matrices U and V gotten from the stochastic SVD decomposition, and then based the framework of alternating nonnegative least squares, we adopt penalty trust region algorithms to construct a new method for NMF. Numerical experiments results demonstrate the high performance of the algorithm.

Keywords: Nonnegative Matrix Factorization, Trust Region, Penalty Function, Randomized SVD

1 Introduction

The concept of Nonnegative Matrix Factorization (NMF) was first proposed by Paatero and Tapper [8] in 1994. Because of its simple decomposition and less storage, NMF has been widely applied in the fields, such as face recognition [14], image analysis [2], signal processing [1] and so on. NMF can be stated as the follows: Given a nonnegative matrix $A \in \mathbb{R}^{m\times n}$, finds two nonnegative matrices $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$ such that:

$$A \approx WH, \tag{1.1}$$

where $r \ll \min(m,n)$. When the Euclidean distance between W and H is applied, problem (1.1) can be rewrote as following optimal problems:

$$\begin{cases} \min F(W,H) = \frac{1}{2}\|A-WH\|_F^2 \\ s.t. \quad W \geq 0, H \geq 0 \end{cases}, \tag{1.2}$$

where $\|\cdot\|_F$ is the Frobenius norm.

In the last decade, there have been many algorithms for solving the problem (1.2). Most of them are based on multiplicative update [6] or alternating nonnegative least squares [10]. The basic idea of alternating nonnegative least squares is to solve the following two subproblems:

$$H^{k+1} = \arg\min_{H\geq 0} F\left(W^k, H\right), \tag{1.3}$$

and

$$W^{k+1} = \arg\min_{W\geq 0} F\left(W, H^{k+1}\right). \tag{1.4}$$

When we apply different ways to solve the subproblem (1.3) and (1.4) above, we can have different methods to solve the original problem based on alternating nonnegative least squares, such as Projected Gradient Methods [7] and Alternating Projected Barzilai-Borwein Methods [4]. In [5] and [9], interior point trust region methods for solving NMF problems have been proposed because of the strong convergence properties of the trust region method [11]. Inspired by the above results, we present a new penalty trust region method based on alternating nonnegative least squares framework. Meanwhile, we apply the random singular value matrix decomposition technique to reduce the size of the matrix used in the subproblem.

This paper is organized as follows. A penalty trust region method for solving subproblems is introduced in Section 2.1. A reduce dimension technique is presented in Section 2.2. The whole structure based on alternating nonnegative least squares framework is listed in Section 2.3. The numerical results and the conclusion of this paper are illustrated in Sections 3 and 4, respectively.

2 Penalty Trust Region Algorithms for NMF

2.1 Penalty Trust Region method for subproblems

We first consider the following nonnegative least squares problem:

$$\begin{cases} \min f(x) = \frac{1}{2}\|Ax-b\|^2 \\ s.t. \quad x \geq 0 \end{cases}, \tag{2.1}$$

where $f(x)$ is a real function defined in $\mathbb{R}^n$. Define

$$c^-(x) = \min(x,0), \tag{2.2}$$

then the nonnegative constraints in problem (2.1) are equivalent to:

$$\left\|c^{-}(x)\right\|_2 = 0. \tag{2.3}$$

We have the penalty function respect to (2.1) as follows:

$$P_\sigma(x) = f(x) + \sigma\left\|c^{-}(x)\right\|_2^2, \tag{2.4}$$

where $\sigma > 0$ is a penalty parameter, next, we get the trust region subproblems corresponding to (2.4),

$$\begin{cases} \min \phi_k(d) = g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k \left\|\left(x_k + d\right)^{-}\right\|_2^2, \\ s.t. \quad \|d\|_2 \le \Delta_k \end{cases} \tag{2.5}$$

where $g_k = g(x_k) = \nabla f(x_k)$, Δ_k is the radius of the trust region, B_k is an approximate Hessian of the Lagrange function in (2.1). Denote

$$P_k(x) = P_{\sigma_k}(x). \tag{2.6}$$

We can apply trust region algorithm in [13] to solve the problem (2.1). Since $f(x)$ and x are continuously differentiable, by the similar proof, we can get the following theorem.

Theorem 1. Assuming that $\{x_k\}$ and $\{B_k\}$ are uniformly bounded, if $\sigma_k = \sigma$ for all large k, then the sequence $\{x_k\}$ is not bounded away from K-T points.

Now, we adopt the method above to solve the matrix problem (1.3) and (1.4). firstly, we rewrite problem (1.3) as the form of (2.1), namely,

$$\begin{cases} \min F(H) = \frac{1}{2}\left\|W^k H - A\right\|_F^2. \\ s.t. \quad H \ge 0 \end{cases} \tag{2.7}$$

In order to use the same algorithm framework to solve problem (1.4), we take the following transformation to make (1.3) and (1.4) have same structure,

$$\begin{cases} \min F\left(W^T\right) = \frac{1}{2}\left\|\left(H^{k+1}\right)^T W^T - A^T\right\|_F^2. \\ s.t. \quad W \ge 0 \end{cases} \tag{2.8}$$

Therefore, we only consider the problem (2.7), Denote

$$\left(c^{-}(H)\right)_{ij} = \min\left(H_{ij}, 0\right), \tag{2.9}$$

then constraints condition $H \ge 0$ is equivalent to:

$$\left\|c^{-}(H)\right\|_F = 0. \tag{2.10}$$

The corresponding subproblem is:

$$\begin{cases} \min \phi_k(d) = \langle g_k, d\rangle + \frac{1}{2}\langle d, B_k d\rangle + \sigma_k \left\|\left(H_k + d\right)^{-}\right\|_F^2, \\ s.t. \quad \|d\|_F \le \Delta_k \end{cases} \tag{2.11}$$

where $g_k = g(H_k) = \nabla F(H_k) = (W^k)^T (W^k H^k - A)$, $\langle \cdot,\cdot \rangle$ is the sum of the component-wise product of two matrices. Let $P_\sigma(H) = F(H) + \sigma \left\| c^-(H) \right\|_F^2$. The matrix form algorithm for solving problem (2.7) is given below.

Algorithm 1.

Step 1 Given $H_1 \in \mathbb{R}^{r\times n}$, $\Delta_1 > 0$, $B_1 \in \mathbb{R}^{r\times r}$ symmetric, $\delta_1 > 0$, $\sigma_1 > 0$, $k = 1$.

Step 2 Solve subproblem (2.11), giving s_k; if $s_k = 0$ then stop;

Step 3 Calculate

$$r_k = \frac{P_k(H_k) - P_k(H_k + s_k)}{\phi_k(0) - \phi_k(s_k)}; \tag{2.12}$$

If $r_k > 0$, then go to Step 4; Otherwise, $\Delta_{k+1} = \left\| s_k \right\|_F / 4$; $H_{k+1} = H_k$; $k = k+1$; go to Step 2;

Step 4 $H_{k+1} = H_k + s_k$;

$$\Delta_{k+1} = \begin{cases} \max\left(2\Delta_k, 4\left\| s_k \right\|_F\right), & r_k > 0.9 \\ \Delta_k, & 0.1 \le r_k \le 0.9; \\ \min\left(\Delta_k / 4, \left\| s_k \right\|_F / 2\right), & r_k < 0.1 \end{cases} \tag{2.13}$$

Generate B_{k+1}.

Step 5 if

$$\phi_k(0) - \phi_k(s_k) < \delta_k \sigma_k \min\left(\Delta_k, \left\| c^-(H_k) \right\|_F\right), \tag{2.14}$$

$\sigma_{k+1} = 2\sigma_k$, $\delta_{k+1} = \delta_k / 4$; Otherwise, $\sigma_{k+1} = \sigma_k$; $\delta_{k+1} = \delta_k$; $k = k+1$, go to Step 2.

2.2 Randomized SVD for Matrix Factorization

With the increase of the scale of the problem, we need to reduce the dimension of the original matrix in order to accelerate the decomposition speed and save the calculation cost. In [12], authors proposed a nonnegative matrix factorization method with random projections, derived a smaller matrix from the original input matrix. In this paper, we apply the results of a two-stage approach Random SVD algorithm [3] to reduce the dimension of the matrix.

A two-stage approach [3] solves the following problem: Given $A \in \mathbb{R}^{m\times n}$, a target rank k, finds matrix factors $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$ such that

$$A \approx UDV^T, \tag{2.15}$$

where U and V are orthonormal, D is diagonal. The following is the algorithm.

Algorithm 2. (Basic Randomized SVD, RSVD)

Input: A matrix $A \in \mathbb{R}^{m\times n}$, a target rank k, an over-sampling parameter p,

```
    $k+p \ll \min(m,n)$.
Output: Matrices $U$, $D$, and $V$.
Stage A:
    (1) Form a Gaussian random matrix $G \in \mathbb{R}^{n\times(k+p)}$.
    (2) Form the sample matrix $Y = AG$.
    (3) Orthonormalize the columns of the sample matrix $Q = \operatorname{orth}(Y)$.
Stage B:
    (4) Form $Q^T A = B \in \mathbb{R}^{(k+p)\times n}$.
    (5) Form the SVD of the small matrix $B$: $B = \hat{U}DV^T$.
    (6) Form $U = Q\hat{U}$.
```

It is easy to see that all the error incurred by the RSVD algorithm is in Stage A, and we also have

$$\left\|A-UDV^T\right\| = \left\|A-QQ^TA\right\|. \tag{2.16}$$

We next analyse the error in the process of applying U and V to reduce the dimension of the original matrix. According to (1.2), We denote

$$J(A) = \|A-WH\|_F^2. \tag{2.17}$$

Decompose the matrix A by Algorithm 2, i.e., $A \approx UDV^T$. We have

$$\hat{J}(A) = \left\|UDV^T - WH\right\|_F^2. \tag{2.18}$$

Since the Frobenius norm is a unitarily invariant norm, then

$$\left\|UDV^T - WH\right\|_F = \left\|DV^T - U^TWH\right\|_F = \|UD - WHV\|_F = \left\|D - U^TWHV\right\|_F. \tag{2.19}$$

From (2.17) and (2.18),

$$\left|J(A) - \hat{J}(A)\right| \le \left(\|A-WH\|_F + \left\|UDV^T - WH\right\|_F\right)\left\|A-UDV^T\right\|_F. \tag{2.20}$$

We assume that $\left\|UDV^T - WH\right\|_F \le \epsilon_1$, where ϵ_1 is a positive constant. According to the conclusions in reference [3], (2.16) and (2.20), we have the following theorem.

Theorem 2. Suppose that $A \in \mathbb{R}^{m\times n}$ is a real matrix with singular values $\sigma_1 \ge \sigma_2 \ge \sigma_3 \ge \ldots$ Choose a target rank $k \ge 2$ and an over-sampling parameter $p \ge 4$, where $k+p \le \min(m,n)$. Draw a standard Gaussian matrix $G \in \mathbb{R}^{n\times(k+p)}$, and construct the sample matrix $Y = AG$. For all $u,t \ge 1$,

$$\left|J(A) - \hat{J}(A)\right| \le 2\epsilon_1\epsilon_2 + \epsilon_2^2, \tag{2.21}$$

with failure probability at most $5t^{-p} + 2e^{-u^2/2}$, where

$$\epsilon_2 = \left(1 + t\sqrt{12k/p}\right)\left(\sum_{j>k}\sigma_j^2\right)^{1/2} + ut\frac{e\sqrt{k+p}}{p+1}\sigma_{k+1}.$$

Remark. If Algorithm 1 convergence, there must exists a constant $\epsilon > 0$ such that $\left\| UDV^T - WH \right\|_F \le \epsilon$.

2.3 Penalty Trust Region Algorithms with reduction technique applied into NMF

In this subsection, we present a new method which summarize the procedures above.

Algorithm 3. Randomized Penalty Trust Region method (RPTR)

Step 1 Given $A \in \mathbb{R}^{m\times n}$, Decompose the matrix A by **Algorithm 2**, $A \approx UDV^T$.

Step 2 Use **Algorithm 1** to solve the following subproblems:

$$H^{k+1} = \arg\min_{H\ge 0} F\left(W^k, H\right) = \frac{1}{2}\left\| DV^T - U^T W^k H \right\|_F^2, \tag{2.22}$$

and

$$W^{k+1} = \arg\min_{W\ge 0} F\left(W, H^{k+1}\right) = \frac{1}{2}\left\| UD - WH^{k+1}V \right\|_F^2. \tag{2.23}$$

Step 3 Output W and H.

3 Numerical experiments

We compare the proposed RPTR methods with Multiplicative Update Methods [6] (MU), Projected Gradient Methods [7] (PG) and Alternating Projected Barzilai-Borwein Methods [4] (APBB2). All implementations were performed on MATLAB.

In Algorithm 1, we use the Matlab build-in function *fmincon* to solve the subproblem (2.11). The initial value B_1 is set to I and B_{k+1} is updated by BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{3.1}$$

where $y_k = g\left(x_k + s_k\right) - g\left(x_k\right)$, $\Delta_1 = 10$, $\sigma_1 = 10$, $\delta_1 = 0.01$. The stopping criterion in Step 2 should be $\left\| s_k \right\|_F \le \epsilon$ for some small positive tolerance number ϵ in the practical implementation of the algorithm.

Let $\nabla^p F\left(W^k, H^k\right)$ be the projected gradient, the stopping criterion in Algorithm 3 is set as

$$\left\| \nabla^p F\left(W^k, H^k\right) \right\|_F \le \epsilon \left\| \nabla F\left(W^0, H^0\right) \right\|_F. \tag{3.2}$$

Test 1. We consider the problem with size 25×25, 50×50 and 100×100. The matrix is randomly generated by the uniform distribution: $A = \text{rand}\left(m, n\right)$. The initial $\left(W^0, H^0\right)$ is constructed by the same way. The values of $k+p$ are in the following order: 15, 25, 75. We try 10 different sample matrices for the same size, and report the average results. All the methods share the same initial point. We set value of $\epsilon = \frac{\left\|\nabla F\left(W^k, H^k\right)\right\|_F}{\left\|\nabla F\left(W^0, H^0\right)\right\|_F}$ to be $\epsilon = 10^{-2}$ and 10^{-3} in order to investigate the results. Let *iter* denote the number of external iterations and *niter* denote the number of internal iterations, $e = \frac{\left\|A - W^k H^k\right\|_F^2}{\left\|A - W^0 H^0\right\|_F^2}$ is the relative error, (m, n, r) is the size of matrix and the dimension of decomposition. We list the results in the following Table 1.

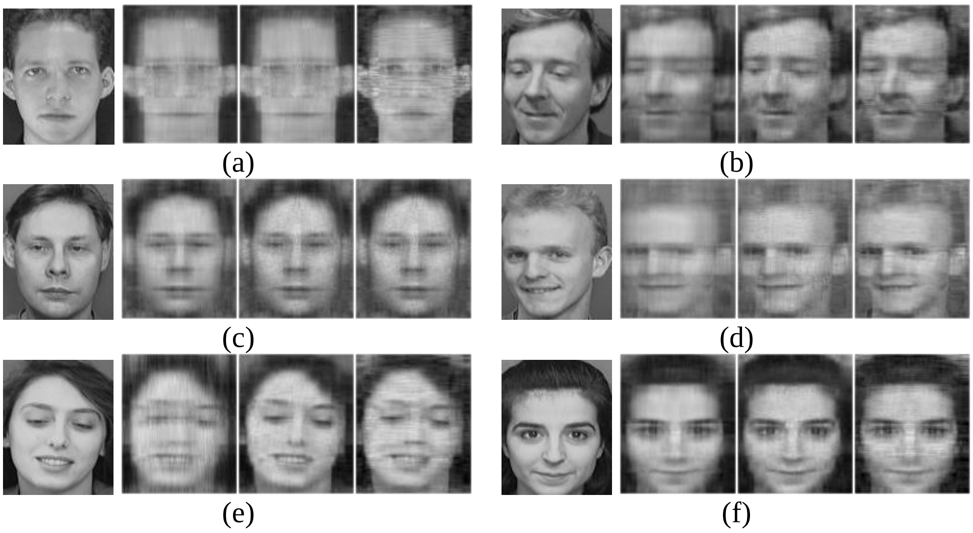
Table 1: Results for random matrices problems

m,n,r		**iter**		**niter**		**e**		ϵ	
		1.0E-02	1.0E-03	1.0E-02	1.0E-03	1.0E-02	1.0E-03	1.0E-02	1.0E-03
25,25,5	RPTR	2	4.1	18.7	56.3	0.0650	0.0566	2.3718E-03	6.9299E-04
	PG	4.3	13.9	59.6	299.1	0.0599	0.0502	7.5900E-03	4.9791E-04
	APBB2	3.1	4.2	26.7	118.5	0.0922	0.1265	7.5555E-03	9.1039E-04
	MU	74	974			0.0517	0.0492	9.2900E-03	9.5323E-04
25,25,6	RPTR	2	6.3	21.7	105.4	0.0377	0.0290	2.6800E-03	7.2912E-04
	PG	4.1	11.6	56.1	240.4	0.0371	0.0274	6.4700E-03	7.4984E-04
	APBB2	2.4	5.7	15	139.8	0.0586	0.0766	6.4700E-03	8.5831E-04
	MU	57	863			0.0284	0.0259	9.4200E-03	9.3915E-04
50,50,5	RPTR	2	2.6	21.5	34.2	0.0800	0.0782	1.1087E-03	6.3933E-04
	PG	2.8	4.9	17	71.4	0.0849	0.0727	9.1000E-03	8.8632E-04
	APBB2	2.2	5.1	12.2	158.1	0.1007	0.1230	6.0700E-03	9.1243E-04
	MU	37	921			0.0748	0.0694	9.3600E-03	9.5935E-04
50,50,6	RPTR	2	2.9	21.2	40.8	0.0486	0.0461	1.0967E-03	6.7649E-04
	PG	2.6	5.2	14.4	62.2	0.0533	0.0431	7.2800E-03	8.3078E-04
	APBB2	2	6	6.1	134.3	0.0645	0.0791	7.2700E-03	8.9024E-04
	MU	31	700			0.0460	0.0408	9.2300E-03	9.5989E-04
100,100,5	RPTR	2	2	21.4	21.4	0.0860	0.0860	4.9236E-05	4.9236E-05
	PG	4	9	44.7	148.9	0.0870	0.0812	8.4300E-03	4.2174E-04
	APBB2	2	5.6	7	123	0.1030	0.1118	4.9500E-03	8.2759E-04
	MU	18	601			0.0882	0.0803	8.5400E-03	9.6978E-04
100,100,6	RPTR	2	2	21.9	21.9	0.0558	0.0558	9.5545E-05	9.5545E-05
	PG	3.9	6.5	31.7	96.4	0.0563	0.0526	4.1881E-03	5.0929E-04
	APBB2	2	5.7	6.1	107.7	0.0687	0.0752	6.1400E-03	9.2880E-04
	MU	10	621			0.0596	0.0514	8.4500E-03	9.6815E-04

From the experimental results in the Table 1, we can see that the number of external and internal iterations of PRTR methods is less than that of the other three algorithms in most cases for the same stopping criterion. The advantage becomes more obvious with the improvement of precision and the enlargement of matrix size. It demonstrates that the algorithm is expected to improve the efficiency of matrix factorization. Furthermore, the relative errors are a little difference among PRTR methods, MU methods and PG method, but are obviously bigger than other three methods in APBB2 methods.

Test 2. We select six pictures randomly from the ORL Faces Database. We apply our method, Multiplicative Update Methods [6] and Projected Gradient Methods [7] to solve the image decomposition, and compare the corresponding reconstructed matrices in Table 2. The dimension of decomposition in the test of three methods is $r = 15$, the stopping criterion is $\epsilon = 10^{-3}$.

Table 2: From left to right, original image, reconstructed images of PG, MU, PRTR



(a)

(b)

(c)

(d)

(e)

(f)

The face images can actually be understood as a weighted linear combination of base images for NMF. The base image is generally composed of local features of each part of the face. By comparing, the reconstructed images obtained by PRTR methods have more distinct features in the main parts of face, like eyes, mouth and nose, etc. In addition, the number of iterations of PRTR methods is lesser for the same stopping criterion in image decomposition.

4 Conclusions

This paper constructs the trust region model by introducing penalty functions, and proposes a new algorithm for NMF based on the trust region methods. The numerical experiments results demonstrate the high performance of the new algorithm. However, the algorithm still needs to be improved. For example, it is convenient that use algorithm *fmincon* to solve the subproblem (2.11), but it does not work very well on large-scale matrix factorization. In the following research, we can try to find a more efficient algorithm to make further improvement on the efficiency of the whole algorithm.

Acknowledgements. The work was supported by the National Science Foundation of China (Grant nos. 11671205).

References

[1] I. Buciu, Nonnegative Matrix Factorization, A New Tool for Feature Extraction: Theory and Applications, *International Journal of Computers, Communications and Control*, **3** (2008).

[2] A. Cichocki, H. Lee, Y.-D. Kim, C. Seungjin, Non-negative matrix factorization with α-divergence, *Pattern Recognition Letters*, **29** (2008), 1433-1440. https://doi.org/10.1016/j.patrec.2008.02.016

[3] N. Halko, P.-G. Martinsson, J. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, *SIAM Review*, **53** (2001), 217-288. https://doi.org/10.1137/090771806

[4] L.X. Han, M. Neumann, U. Prasad, Alternating projected Barzilai-Borwein methods for Nonnegative Matrix Factorization, *Electronic Transactions on Numerical Analysis ETNA*, **36** (2010), 54-82.

[5] J.J. Jiang, H.B. Zhang, S. Yu, An interior point trust region method for nonnegative matrix factorization, *Neurocomputing*, **97** (2012), 309–316. https://doi.org/10.1016/j.neucom.2012.05.008

[6] D. Lee, H. Seung, Algorithms for Non-negative Matrix Factorization, *Adv. Neural Inform. Process. Syst.*, **13** (2001).

[7] Lin, C.J., Projected Gradient Methods for Nonnegative Matrix Factorization, *Neural Computation*, **19** (2007), 2756-2779. https://doi.org/10.1162/neco.2007.19.10.2756

[8] P. Paatero, U. Tapper, Positive Matrix Factorization: A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values. *Environmetrics*, **5** (1994), 111-126. https://doi.org/10.1002/env.3170050203

[9] M. Rojas, T. Steihaug, An Interior-Point Trust-Region-Based Method for Large-Scale Nonnegative Regularization, *Inverse Problems*, **18** (2002), 1291-1307. https://doi.org/10.1088/0266-5611/18/5/305

[10] D.K. Smith, D.P. Bertsekas, Nonlinear Programming, *The Journal of the Operational Research Society*, **48** (1997), 334. https://doi.org/10.1057/palgrave.jors.2600425

[11] W.Y. Sun, C.X. Xu, D.T. Zhu, *Optimization Methods*, Higher Education Press, 2010.

[12] F. Wang, P. Li, Ping, Efficient Nonnegative Matrix Factorization with Random Projections, *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, 2010, 281-292. https://doi.org/10.1137/1.9781611972801.25

[13] Y.X. Yuan, On the convergence of a new trust region algorithm, *Numer. Math.*, **70** (1995), 515-539. https://doi.org/10.1007/s002110050133

[14] S. Zafeiriou, A. Tefas, I. Buciu, I. Pitas, Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification, *IEEE Transactions on Neural Networks / a publication of the IEEE Neural Networks Council*, **17** (2006), 683-95. https://doi.org/10.1109/tnn.2006.873291

Received: December 11, 2020; Published: December 25, 2020