STOCHASTIC VARIANCE REDUCED MULTIPLICATIVE UPDATE FOR NONNEGATIVE MATRIX FACTORIZATION

Hiroyuki Kasai

The University of Electro-Communications, Tokyo, JAPAN

ABSTRACT
Nonnegative matrix factorization (NMF), a dimensionality reduction and factor analysis method, is a special case in which factor matrices have low-rank nonnegative constraints. Considering the stochastic learning in NMF, we specifically address the multiplicative update (MU) rule, which is the most popular, but which has slow convergence property. This present paper introduces on the stochastic MU rule a variance-reduced technique of stochastic gradient. Numerical comparisons suggest that our proposed algorithms robustly outperform state-of-the-art algorithms across different synthetic and real-world datasets.

Index Terms— nonnegative matrix factorization, multiplicative update, stochastic gradient, variance reduction

1. INTRODUCTION
Superior performance of nonnegative matrix factorization (NMF) has been achieved in many technical fields. NMF approximates a nonnegative matrix \( V \) as a product of two nonnegative matrices \( W \) and \( H \). Given \( V \in \mathbb{R}^{F \times N} \), NMF requires factorization of the form \( V \approx WH \), where \( W \in \mathbb{R}^{F \times K} \) and \( H \in \mathbb{R}^{K \times N} \) are nonnegative factor matrices. \( K \) is usually chosen such that \( K \leq \min\{F,N\} \), that is, \( V \) is approximated in the two low-rank matrices. This problem is formulated as a constrained minimization problem in terms of the Euclidean distance as

\[
\min_{WH} \quad \frac{1}{2} \|V - WH\|_F^2 = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} \|v_n - Wh_n\|_2^2, \\
\text{s.t.} \quad [W]_{f,k} \geq 0, [H]_{k,n} \geq 0, \quad \forall f, n, k,
\]  

(1)

where \( V = [v_1, \ldots, v_N] \) and \( H = [h_1, \ldots, h_N] \). \( [A]_{i,j} \) is the \((i,j)\)-th entry of \( A \). The non-negativity of \( V \) enables us to interpret the meanings of the obtained matrices \( W \) and \( H \) well. This interpretation produces a broad range of applications in machine learning and signal processing such as text mining, image processing, and data clustering, to name a few. However, because problem (1) is a non-convex optimization problem, finding its global minimum is NP-hard. For this problem, Lee and Seung proposed a simple but effective calculation algorithm [1] as

\[
H \leftarrow H \circ \frac{W^T V}{W^T WH}, \quad W \leftarrow W \circ \frac{V H^T}{WH^T},
\]

(2)

where \( \circ \) (resp. \( \odot \)) denotes the component-wise product (resp. division) of matrices, which finds a local optimal solution of (1). This rule is designated as the multiplicative update (MU) rule because a new estimate is represented as the product of a current estimate and some factor. The global convergence to a stationary point is guaranteed under slightly modified update rules or constraints [2, 3]. Nevertheless, many efficient algorithms have been developed because the MU rule is accompanied by slow convergence [4, 5, 6, 7]. Furthermore, considering big data, an online learning algorithm is preferred in terms of the computational burden and the memory consumption. Designating the former algorithms as batch-NMF, this online-NMF has been investigated actively in several studies [8, 9, 10, 11]. Its robust variant has also been assessed [12]. However, they still exhibit slow convergence. Stochastic gradient descent (SGD) [13] has become the method of choice for solving big data optimization problems. Although it is beneficial because of the low and constant cost per iteration independent of \( N \), the convergence rate of SGD is also slower than that of full GD even for the strongly convex case. For this issue, various variance reduction (VR) approaches that have been proposed recently have achieved superior convergence rates in convex and non-convex functions.

This paper presents a proposal of a novel stochastic multiplicative update with the VR technique: SVRMU. The paper also explains extension of SVRMU to the accelerated variant (SVRMU-ACC), and the robust variant (R-SVRMU) for outliers. The paper is organized as follows. Section 2 presents details of the variance reduction algorithm in stochastic gradient. Section 3 presents the proposed stochastic variance reduced multiplicative update (SVRMU). Section 4 provides a convergence analysis. Two extensions are detailed in Section 5. In Section 6, exhaustive comparisons suggest that our proposed SVRMU algorithms robustly outperform state-of-the-art algorithms across different synthetic and real-world datasets. It is noteworthy that the discussion presented here is applicable to other distance functions than the Euclidean distance. The Matlab codes are available at https://github.com/hiroyuki-kasai.
2. VARIANCE REDUCTION ALGORITHM IN
STOCHASTIC GRADIENT

An algorithm designated to solve the problem (1) without
nonnegative constraints is begin eagerly sought in the ma-
cine learning field. When designating W and H as w, and
designating the rightmost inner term of the cost func-
tion (1) as \( f_i(w) \), respectively, full gradient decent (GD)
with a stepsize \( \eta \) is the most straightforward approach as
\( w_{t+1} = w_t - \eta \nabla f(w_t) \), where \( \nabla f(w_t) \) corresponds to the
gradient \( g_t \). However, this is expensive especially when \( N \)
is extremely large. A popular and effective alternative is a
stochastic gradient descent (SGD). It updates \( w_t \) as \( w_{t+1} = w_t - \eta \nabla f_n(w_t) \), and assumes an unbiased
estimator of the full gradient as \( \mathbb{E}_n[\nabla f_n(w_t)] = \nabla f(w_t) \).

Apparently, the calculation cost per iteration is independent
of \( N \). Mini-batch SGD uses \( g_t = 1/|S_t| \sum_{n \in S_t} \nabla f_n(w_t) \),
where \( S_t \) is the set of samples of size \( |S_t| \). However, because
SGD requires a diminishing stepsize algorithm to guarantee
the convergence, SGD suffers from a slow convergence rate.

To accelerate this rate, the variance reduction (VR) tech-
niques [14, 15, 16, 17, 18, 19] explicitly or implicitly ex-
agon the problem setting is the following: we assume that
\( \mathbf{h}_n \) is extremely large. A popular and effective alternative is a
stochastic gradient descent (SGD). It updates \( w_t \) as \( w_{t+1} = w_t - \eta \nabla f_n(w_t) \), and assumes an unbiased
estimator of the full gradient as \( \mathbb{E}_n[\nabla f_n(w_t)] = \nabla f(w_t) \).

\[
\begin{align*}
2 & \text{ Calculate the components of the full gradient } \nabla w = \langle \nabla f \rangle, \text{ as in (4). Hereinafter, } k \text{ is used instead of } n_k \text{ for notation simplicity. Then, we update }
W_t^s \text{ with an appropriate stepsize } S_t^s \text{ as shown below.}
\end{align*}
\]

\[
W_{t+1} = W_t - S_t^s \left[ \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} \right) - \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} + \frac{W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T}}{N} \right) \right]
\]

Algorithm 1 Stochastic variance reduction multiplicative update (SVRMU)

\[
\begin{align*}
1: & \text{ Initialize } W^s \text{ and } h^s. \\
2: & \text{ for } s = 0, 1, \ldots \text{ do } \\
3: & \text{ Calculate the components of the full gradient } \nabla w = \langle \nabla f \rangle, \text{ as in (4). Hereinafter, } k \text{ is used instead of } n_k \text{ for notation simplicity. Then, we update }
W_t^s \text{ with an appropriate stepsize } S_t^s \text{ as shown below.}
\end{align*}
\]

\[
W_{t+1} = W_t - S_t^s \left[ \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} \right) - \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} + \frac{W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T}}{N} \right) \right]
\]

\[
\frac{\alpha W^s(X_t^s h_{n_k}^T) \in \mathbb{R}^{F \times K}}{W_t^s - S_t^s \left[ \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} \right) - \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} + \frac{W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T}}{N} \right) \right]
\]

3. STOCHASTIC VARIANCE REDUCED
MULTIPLICATIVE UPDATE (SVRMU)

This section first describes the stochastic multiplicative update, designated as SMU. Then it details the proposed
stochastic variance-reduced MU algorithm, i.e., SVMU.
The problem setting is the following: we assume that \( \mathbf{h}_n \in \mathbb{R}^N \) of \( \mathbf{V} \), i.e., \( \mathbf{h}_n \), is selected at \( t \)-th iteration
uniformly at random. \( \mathbf{h}_n \) and \( \mathbf{W} \) are updated simultaneously by extending (2) as

\[
\begin{align*}
\text{Algorithm 1 Stochastic variance reduction multiplicative update (SVRMU)}
\end{align*}
\]

\[
\frac{\alpha W^s(X_t^s h_{n_k}^T) \in \mathbb{R}^{F \times K}}{W_t^s - S_t^s \left[ \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} \right) - \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} + \frac{W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T}}{N} \right) \right]
\]

\[
\frac{\alpha W^s(X_t^s h_{n_k}^T) \in \mathbb{R}^{F \times K}}{W_t^s - S_t^s \left[ \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} \right) - \left( W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T} + \frac{W_{t} h_{t} h_{t}^{T} - v_{t} h_{t}^{T}}{N} \right) \right]
\]
We also denote $P_t^s \in \mathbb{R}_{+}^{F \times K}$ as
\[
P_t^s = v_{h_k^t} + \mathbb{W}_h^s (h_{k}^t)^T + \mathbb{V}(H_h^s)^T N.
\]
When $S_t^s = \alpha_t^2 P_t^s / Q_t^s$ with the stepsize ratio $\alpha_t^2$, the update rule in (5) is reformulated as presented below.
\[
W_{t+1}^s = W_t^s - \alpha_t W_t^s / Q_t^s \odot (Q_t^s - P_t^s).
\]

The overall algorithm is summarized in Algorithm 1. Additionally, the straightforward extension to the mini-batch variant of Algorithm 1 is defined as
\[
W_{t+1}^s = W_t^s - S_t^s \odot \left( \frac{W_t^s h_k^t + v_{h_k^t} (h_{k}^t)^T + \mathbb{W}_h^s (H_h^s)^T}{b} \right) - \left( \frac{v_{h_k^t} (h_{k}^t)^T + \mathbb{W}_h^s (h_{k}^t)^T + \mathbb{V}(H_h^s)^T}{b} \right),
\]
where $b (\leq N)$ is the mini-batch size. $Q_t^s$ and $P_t^s$ in (6) are modified accordingly.

4. CONVERGENCE ANALYSIS

The convergence analysis is similar to [21, 22], but is different because of the update rule in (5). More specifically, denoting the rightmost term in (1) as $l(h_n, W) := \frac{1}{2} \| v_n - \mathbb{W} h_n \|_2^2$, we define the empirical cost $f_N(h_n, W) := \frac{1}{N} \sum_{n=1}^{N} l(h_n, W)$. We also define $f_N(W) := \frac{1}{N} \sum_{n=1}^{N} l(h_n, W)$, where $h_n$ is already calculated during the previous steps. We now consider the expected cost $f(h_n, W) := E_W[l(h_n, W)] = \lim_{N \rightarrow \infty} f_N(h_n, W)$, which is taken with respect to the distribution $P(v)$ of the samples. Our interest is usually in the minimization of the expected cost $f(h_n, W)$ almost surely (a.s.) instead of the empirical cost $f_N(h_n, W)$. To this end, the convergence analysis first shows that $f_N(h_n, W) \rightarrow f(h_n, W)$ converge a.s. to zero, where $f_N(W)$ acts as a surrogate function for $f_N(h_n, W)$. For this proof, we show that $f_N(W)$ converges a.s. under the modified update rule in (5). Here, the stepsize ratio $\alpha_t^2$ plays a crucial role in generating a diminishing sequence of $S_t^s$ to guarantee its convergence. After showing the convergence of $f_N(h_n, W)$, we finally obtain below:

Theorem 4.1. Assume that $\{v_n\}_{n=1}^{\infty}$ are i.i.d. random processes, and bounded. Iterates of $W_t^s$ for $0 \leq t \leq m_s - 1$ and $0 \leq s$ are compact. The initial $W_0^s$ is nonnegative and has a full column rank. $f_N(W)$ is positive definite and strictly convex. $\alpha_t^2$ generates a diminishing stepsize of $S_t^s$. Then, the iterates $W_t^s$ produced by Algorithm 1 asymptotically coincide with the stationary points of the minimization problem of $f(h_n, W)$.

5. EXTENSIONS OF SVRMU

This section proposes two variants of SVRMU.

5.1. Accelerated SVRMU (SVRMU-ACC)

Close examination of the update rule of $h_k$ and $W_t^s$ reveals that, whereas the latter requires $3FK + 2FN$ because of the dominant calculation of the component-wise product of $W_t^s$ at the last step, the former requires only $3FK + 2K$, which is much lower than that of the latter because of $K \ll \{F, N\}$. Therefore, we can repeat the calculation of $h_k$ several times, which corresponds to Step 7 in Algorithm 1, before the computation of $W_t^s$. Although a similar strategy is also proposed for the batch-based MU [6], the proposed one differs because of the different update rule. The noteworthy point is the stopping criteria, which are (i) the maximum iteration number $L$, and (ii) the dynamic stop criteria. The former specifically examines the ratio of the calculation complexity between $W_t^s$ and $h_k$. We calculate $L = \max\{\lceil \frac{3FK + 2FN}{3FK + 2K} \rceil, 1\}$, where $0 \leq \beta \leq 1$. Regarding the dynamic stop criteria, the process stops when the change between $l$-th $h_k^{(l)}$ and $(l - 1)$-th $h_k^{(l-1)}$ falls below the predefined ratio $\epsilon$ of the difference from the initial value $h_k^{(0)}$. The algorithm is presented as Algorithm 2.

Algorithm 2 Repetitive calculation algorithm of $h_k$.

Require: $h_k^{(0)}, (W_t^s)_{t=0}^{\infty}, v_k$, and the ratio $\epsilon$.
1: Set $h_k^{(0)} = h_k$.
2: for $l = 1, 2, \ldots, L$ do
3: \hspace{0.5cm} Calculate $h_k = h_k \odot [(W_t^s)^T v_k / (W_t^s)^T W_t^s h_k)$.
4: \hspace{0.5cm} \textbf{if} $\| h_k^{(l)} - h_k^{(l-1)} \|_F < \epsilon \| h_k^{(l)} - h_k^{(0)} \|_F \textbf{ then}
5: \hspace{1cm} break.
6: \hspace{1cm} end if
7: \hspace{0.5cm} end for
8: Return $h_k = h_k^{(l)}$.

5.2. Robust SVRMU (RSVRMU)

The outlier in $V$ causes remarkable degradation of the approximation of $V$. To address this issue, the robust batch-NMF [23] and the robust online-NMF [12] have been proposed. This extension also tackles the same problem within the SVRMU framework. Given the outlier matrix $R = [r_1, \ldots, r_N] \in \mathbb{R}_{+}^{F \times N}$, the robust variant seeks $V \approx WH + R$, of which minimization problem is formulated as
\[
\min_{W, H, R} \frac{1}{N} \sum_{n=1}^{N} \| v_n - \mathbb{W} h_n - r_n \|_2^2 + \lambda \| r_n \|_1,
\]
s.t. $[W]_{f,k} \geq 0, h_n \geq 0, r_n \geq 0, \forall f, n, k,$
where $\lambda > 0$ is the regularization parameter, and $\| \cdot \|_1$ is the $\ell_1$-norm. For this problem, the update rule (5) is redefined as

$$w^s_{t+1} = w^s_t - s^T \bigg[ \frac{(w^s_t h_k + r_k) h^T_k + v_k (h^s_k)^T}{\sum h^T_k W^s_t h_k + (w^s_t)^T r_t} \bigg] \bigg[ (w^s_t h_k + r_k) h^T_k + v_k (h^s_k)^T + \frac{v (h^T_k)^T}{N} \bigg].$$

Accordingly, we respectively calculate as

$$h_k \leftarrow h_k - \frac{(w^T_t)^T v_k}{w^T_t h_k + (w^T_t)^T r_t}$$

$$r_k \leftarrow r_k - \frac{w^T_t h_k + r_k + \Lambda_{F \times 1}}{w^T_t h_k + r_k}.$$

6. NUMERICAL EXPERIMENTS

This section demonstrates the effectiveness of SVRMU by comparing with the state-of-the-art online algorithms for NMF. We implemented all of the algorithms in Matlab.

6.1. Convergence behavior under clear synthetic data

The element $[w_{o}]_{f,n}$ of the ground-truth $w_{o} \in \mathbb{R}^{F \times K_{o}}$ is generated from a Gaussian distribution with a mean of zero and variance $1/\sqrt{K_{o}}$ for any $(f, n)$, where $K_{o}$ is the ground-truth rank dimension. Similarly, we generate $h_{o} \in \mathbb{R}^{K_{o} \times N}$. Then, the clean data $v_{o}$ are created as $v_{o} = P_{V}(w_{o} h_{o})$, where $V = [0, 1]^{F \times N}$, and $P_{V}$ is the normalization projector [12]. We set $(F, N, K_{o}, b) = (300, 1000, 10, 100)$. The maximum epoch is 500. The following methods are used for comparison: incremental MU (INMF) [8], online MU (ONMF) [12], and ASAG-MU [24]. Our proposed algorithms include SMU and SVRMU in Section 3, and those accelerated variants, i.e., SMU-ACC and SVRMU-ACC, in Section 5.1. Figure 1 presents results of the convergence behavior in terms of the optimality gap, which is calculated using HALS [5] in advance. The figure shows the superior performance of SVRMU in terms of the number of gradients and the time.

6.2. Base representation of face image with outlier

We use the CBCL face dataset, which has 2429 gray-scale images of size $19 \times 19$. The maximum level of the pixel values is set to 50. All pixel values are normalized. We also randomly add entry-wise nonnegative outliers with density $\rho = 0.9$. All outliers are drawn from the i.i.d. from a uniform distribution $U([30, 50])$. $K_{o}$ is fixed to 49. The methods of comparison include ONMF and its robust variant: R-ONMF [12], and the accelerated variant of R-SVRMU. The batch-based variant of R-ONMF (R-NMF) is also evaluated. Figure 2 presents an illustration of the generated 14 basis representations, where it is apparent that R-SVRMU produces better bases than R-ONMF, and gives similar bases as the batch-based R-NMF.

7. CONCLUSIONS

This present paper has proposed a novel stochastic multiplicative update with variance reduction technique: SVRMU. Numerical comparisons suggest that SVRMU robustly outperforms state-of-the-art algorithms across different synthetic and real-world datasets.
8. REFERENCES


