관측행렬에 대한 전처리 Cholesky-Factor Downdating 기법

김 석 일† · 이 충 한†† · 전 중 남†††

요 약

본 논문에서는 다행관측행렬 Z^T를 빠르게 downdating하기 위하여 Z^T의 partial Cholesky factor R_Z를 계산하는 전처리 과정을 거친 후, R_Z에 각각 기존의 GD(Givens Downdating)기법과 HD(Hyperbolic Downdating)기법을 적용한 PGD(Preprocessed GD)기법과 PHD(Preprocessed HD)기법을 제안하였다. $p\times n(p\geq n)$ 크기의 다행관측행렬 Z^T를 downdating하는 데 필요한 시간복잡도는 PGD 및 PHD기법을 이용한 downdating의 경우에 각각 $pn^2+5n^3/6$ 및 $pn^2+n^3/3$ flops이며, 기존의 GD기법 또는 HD기법을 이용한 downdating에서는 각각 $5pn^2/2$과 $2pn^2$ flops이므로 다행관측행렬 Z^T를 partial Cholesky factor R_Z^T로 분할하는 전처리 과정이 downdating알고리즘의 성능을 개선할 수 있음을 보여준다. Sun SPARC/2 시스템에서의 벤치마크 실험 결과도 전처리 과정을 거친 알고리즘의 실행 속도가 전처리 과정을 거치지 않은 알고리즘에 비하여 빠른 결과를 얻었으며, 두 가지 전처리 기법 중에서도 PHD 기법이 PGD 기법보다 시간복잡도 측면에서 우수하였다.

Preprocessed Cholesky-Factor Downdatings for Observation Matrices

Sukil Kim† · Chung-Han Lee†† · Joong-Nam Jeon†††

ABSTRACT

This paper introduces PGD(Preprocessed Givens Downdating) and PHD(Preprocessed Hyperbolic Downdating) algorithms, wherein a multiple-row observation matrix Z^T is factorized into a partial Cholesky factor R_Z, such that $Z^T=Q_Z R_Z$, $Q_Z Q_Z^T=I$, and then R_Z is recursively downdated by using GD(Givens Downdating) and HD(Hyperbolic Downdating), respectively. Time complexities of PGD and PHD algorithms are $pn^2+5n^3/6$ and $pn^2+n^3/3$ flops, respectively, if $p\geq n$, while those of the existing GD and HD are known to be $5pn^2/2$ and $2pn^2$ flops, respectively. This concludes that the factorization of observation matrices, which we call *preprocessing*, would improve the overall performance of the downdating process. Benchmarks on the Sun SPARC/2 system also show that preprocessing would shorten the required downdating times compared to those of downdatings without preprocessing. Furthermore, benchmarks also show that PHD provides better performance than PGD.

이 과제는 한국과학재단의 94년도 목적기초연구과제로 수행되었음.

† 종신회원：충북대학교 컴퓨터과학과 부교수
†† 준 회 원：충북대학교 컴퓨터과학과
††† 종신회원：충북대학교 컴퓨터과학과 조교수

논문접수：1995년 7월 26일, 심사완료：1995년 12월 29일

1. 서 론

신호처리에 적용되는 공통적인 문제는 X가 $m\times n$ $(m>n)$ 행렬이며, s가 $m\times 1$ 벡터, w가 $n\times 1$ 벡터인 선형대수식 $Xw=s$의 최소자승해

$$\min \| s-Xw \|_2 \tag{1}$$

를 계산하는 문제로 귀결된다[1].

X의 rank가 n이라고 하고 확대행렬 $(X\ s)$에 QR-분할을 적용하면,

$$Q^T(X\ s)=\begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix}, \qquad QQ^T=I. \tag{2}$$

따라서 식(1)을 만족하는 해 w는 식(2)를 만족하는 직교변환(orthogonal factorization) Q를 계산하고, 이어서 삼각연립방정식 $Rw=Q^Ts$의 해를 구하는 일련의 과정을 거쳐 계산할 수 있다.

마찬가지로 확대행렬 $(X\ s)$에 관측치 $(z^T\ \sigma)$이 순차적으로 입력되어 첨가되거나, $(\tilde{X}\ \tilde{s})$로부터 $(z^T\ \sigma)$이 순차적으로 제거되는 경우에도 변경된 확대행렬 $(\tilde{X}\ \tilde{s})$로부터

$$\min \| \tilde{s}-\tilde{X}\tilde{w} \|_2 \tag{3}$$

를 만족하는 새로운 해 $\tilde{w}$를 계산하는 방법도 식(2)와 같이 $(\tilde{X}\ \tilde{s})$의 직교변환을 취하여 계산할 수 있다. 여기서 관측치 $(z^T\ \sigma)$이 확대행렬 $(X\ s)$에 첨가되는 updating의 경우에는

$$\tilde{X}=\begin{pmatrix} X \\ z^T \end{pmatrix}, \qquad \tilde{s}=\begin{pmatrix} s \\ \sigma \end{pmatrix}.$$

관측치 $(z^T\ \sigma)$이 확대행렬 $(X\ s)$로부터 제거되는 downdating의 경우에는

$$X=\begin{pmatrix} \tilde{X} \\ z^T \end{pmatrix}, \qquad s=\begin{pmatrix} \tilde{s} \\ \sigma \end{pmatrix}.$$

그런데 행렬 X에 관측치 $(z^T\ \sigma)$가 첨가된 경우나 탈락한 경우, Cholesky factor $\tilde{R}$은 R과 z^T로부터 계산할 수 있다. 즉, updating의 경우에는,

$$\tilde{R}^T\tilde{R}=R^TR+zz^T.$$

마찬가지로 downdating의 경우에도 R과 z^T로부터 새로운 Cholesky factor $\tilde{R}$을 계산할 수 있다. 즉,

$$\tilde{R}^T\tilde{R}=R^TR-zz^T.$$

따라서 rank n인 행렬 X에 하나의 관측치가 첨가되거나 제거되는 문제는 rank 1이 변경된 Cholesky factor R의 updating이나 downdating으로 간주할 수 있다. 따라서 행렬 X에 대한 Cholesky factor R이 존재한다면, 관측치 $(z^T\ \sigma)$가 첨가된 경우나 제거된 경우, X보다는 R을 이용하여 새로운 Cholesky factor $\tilde{R}$을 빠르게 계산할 수 있다. 이러한 방법을 각각 CFU (Cholesky Factor Updating) 및 CFD(Cholesky Factor Downdating)라고 한다[4-6, 8, 13-15].

그 중에서도 CFD는 관측치가 연속적으로 입력되는 신호처리 장치에서 하나의 관측치에 대하여 downdating을 반복적으로 수행하는 경우에 자주 사용되는 알고리즘이다[1]. 그 예로는 speech echo cancellation, speech coding 및 adaptive radar signal processing 등이 있다. 이러한 시스템에서 사용될 수 있는 downdating 기법은 실시간 처리를 위하여 가급적 요구되는 연산의 수가 적어야 하며, 또한 계산 결과로 얻은 해의 정확도가 높아야 한다[1].

단위 시간당 p개의 관측치가 주기적으로 입력되는 시스템에서 식(2)의 QR-분할법을 반복적으로 이용하는 알고리즘은 단위 시간당 처리하는 연산 능력이 $O(pmn^2)$ flops이어야 한다. 이에 비하여 CFD기법[6, 8]은 rank 1 downdating을 반복적으로 수행하는 알고리즘으로, 단위 시간당 처리하는 연산 능력이 $O(pn^2)$ flops이면 충분하다. 따라서 CFD기법을 이용하는 시스템은 QR-분할법을 이용하는 시스템에 비하여 단위 시간당 처리할 수 있는 연산 능력이 $1/m$배인 컴퓨터를 사용할 수 있다.

만일 시스템을 구성하고 있는 컴퓨터의 단위시간당 처리할 수 있는 연산능력이 낮은 경우에는 여러 개의 관측치를 모아서 한 번에 처리하여야 한다. 이

러한 경우에는 여러 개의 관측치를 한번에 처리해야 하며, 이 것을 다행관측치의 downdating이라고 한다 [2, 13].

다행관측행렬의 downdating 기법은 기본적으로 CFD를 반복적으로 수행하는 방법이 사용되어 왔으나, Lee[2] 등은 다행관측행렬을 전처리 함으로써 필요로 하는 단위시간당 처리할 수 있는 연산 능력을 낮출 수 있음을 밝힌바 있다. 즉, p개의 관측치가 블록을 이루는 다행관측행렬 $(Z^T\ \sigma)=\begin{pmatrix} z_1^T & \sigma_1 \\ z_2^T & \sigma_1 \\ \vdots & \vdots \\ z_p^T & \sigma_p \end{pmatrix}$가 행렬 $(X\ s)$로부터 제거된 확대행렬 $(\tilde{X}\ \tilde{s})$은

$$\tilde{X}^T\tilde{X}=R^TR-ZZ^T$$

이므로

$$\tilde{R}^T\tilde{R}=R^TR-ZZ^T. \tag{4}$$

관측행렬 Z^T의 partial Cholesky factor(3.1절 참조)를 R_Z이라고 하면

$$Z^T=Q_ZR_Z,\quad Q_Z^TQ_Z=I, \tag{5}$$

따라서 식(4)로부터

$$\tilde{R}^T\tilde{R}=R^TR-ZZ^T=R^TR-R_Z^TR_Z. \tag{6}$$

즉, 새로운 Cholesky factor $\tilde{R}$은 식(6)과 같이 다행관측행렬의 partial Cholesky factor인 R_Z와 이전의 Cholesky factor R로부터 계산할 수 있다. 이 과정에서 R_Z의 대각원소 아래는 모두 0이므로 CFD 기법을 변형하여 이들 원소에 대한 downdating 계산을 배제하면 시스템이 필요로 하는 연산 횟수가 크게 감소한다.

본 논문에서는 Lee[2]의 연구 결과를 토대로 Givens rotation을 이용한 CFD 기법[4, 6, 15]과 hyperbolic rotation을 이용한 CFD 기법[3, 5, 7]을 수행하기에 앞서 식(5)를 만족하는 다행관측행렬 Z^T의 partial Cholesky factor R_Z를 계산하는 전처리과정을 거쳐 R_Z의 0이 아닌 원소에 대해서만 downdating을 수행하는 빠른 CFD기법을 제안하였다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 rank 1 downdating을 수행하는 CFD기법으로 Givens rotation을 이용하는 GD(Givens Downdating)와 가성직교변환(pseudo-orthogonal)을 이용하는 HD(Hyperbolic Downdating) 알고리즘을 소개하였으며, 제 3장에서는 다행관측행렬 Z^T의 partial Cholesky factor R_Z를 계산하기 위한 기법으로 Householder updating [10, 12]을 소개하고 그 결과로 계산된 partial Cholesky factor의 0이 아닌 원소에 대해서만 downdating이 가능하도록 GD기법과 HD기법을 변형한 PGD(Preprocessed GD) 및 PHD(Preprocessed HD) 기법을 제안하였다. 제 4장에서는 본 논문에서 제안한 PGD와 PHD기법의 시간복잡도를 비교하였으며, Sun SPARC 시스템에서 계산에 필요한 시간을 측정하여 이론적인 시간복잡도와 비교하였다. 마지막으로 제 5장에서는 본 연구의 결과를 제시하였다.

2. 다행관측행렬의 Downdating

단위 시간당 입력되는 p개의 관측치 행벡터 $z_k^T(k=1, 2, \cdots, p)$를 하나의 다행관측행렬 $Z^T=(z_1\ z_2\ \cdots\ z_p)^T$로 구성하고 이를 downdating하는 기존의 CFD 기법은 단위 시간당 rank 1 downdating을 p번 반복적으로 수행한다. 여기서 $m\times n$ 행렬 X_k와 $n\times n$인 상삼각행렬 R_k를 각각 Z^T의 $k-1$ 번째 행 z_{k-1}^T을 rank 1 downdating을 수행한 결과라고 가정한다.

2.1 직교변환을 이용한 Downdating

직교변환을 이용한 downdating은 X_k의 QR-분할의 결과로 얻어진 상삼각행렬 R_k와 관측치 z_k^T로부터 R_{k+1}을 계산하는 방법으로, 그 과정 중에서 Givens rotation이 반복적으로 적용되므로 본 논문에서는 이를 GD(Givens Downdating) 기법[4]이라고 부른다.

관측행렬 Z^T의 k번째 행 z_k^T을 downdating할 때의 관측행렬은

$$X_k=\begin{pmatrix} X_{k+1} \\ z_k^T \end{pmatrix},\qquad X_1\equiv X.$$

이 된다. 또한, $(X_k\ s_k)$의 직교변환을 Q_k라고 하면,

$$(X_k\ s_k) = \begin{pmatrix} X_{k+1} & s_{k+1} \\ z_k^T & \sigma_k \end{pmatrix} = Q_k \begin{pmatrix} R_k & u_k \\ 0 & \rho_k \\ 0 & 0 \end{pmatrix} \tag{7}$$

이다. 따라서 행렬 $E = \begin{pmatrix} 0_{m-1} \\ 1 \end{pmatrix}$을 $(X_k\ s_k)$의 좌측에 포함시키면 다음의 식이 성립한다.

$$Q_k^T(E\ X_k\ s_k)$$
$$= Q_k^T \begin{pmatrix} 0_{m-1} & X_{k+1} & s_{k+1} \\ 1 & z_k^T & \sigma_k \end{pmatrix} = \begin{pmatrix} q_1^k & R_k & u_k \\ \phi^k & 0 & \rho_k \\ q_2^k & 0 & 0 \end{pmatrix}$$

여기서 $q_k^T \equiv ((q_1^k)^T\ \phi^T (q_2^k)^T)$는 Q_k의 마지막 행으로 구성된 행렬이다. 따라서 q_k^T의 j번째 원소를 0으로 만들면서 q_k^T의 마지막(m번째) 원소를 갱신하기 위한 평면(j, m)간의 Givens rotation을 $G_j (j=1, 2, \cdots, m-1)$라고 하면 $(z_k^T\ \sigma_k)$를 downdating하는 직교변환 U_k는 다음과 같이 계산할 수 있다. 즉,

$$U_k \begin{pmatrix} q_1^k & R_k & u_k \\ \phi^k & 0 & \rho_k \\ q_2^k & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & R_{k+1} & u_{k+1} \\ 0 & 0 & \rho_{k-1} \\ 1 & z_k^T & \sigma_k \end{pmatrix}$$

$$U_k = G_1\ G_2 \cdots G_{m-1}. \tag{8}$$

식(8)에서 U_k를 계산하기 위해서는 m회의 Givens rotation이 필요하나, q_2^k의 우측에 있는 원소가 모두 0이므로 실제로는 n회의 rotation이 필요하다[6, 8, 14]. 따라서 (그림 1)의 알고리즘과 같이 rank 1 downdating을 z_1^T, z_2^T, $\cdots$, z_p^T에 대하여 순차적으로 적용하면, 다행관측행렬 Z^T에 대한 downdating 결과를 얻을 수 있다.

(그림 1)에서 downdating이 종료되면 다행관측행렬 Z^T에 의하여 갱신된 Cholesky factor는 $\tilde{R} \equiv R_{p+1}$이다. (그림 1)에서 GD기법은 전체적으로 pn회의 Givens rotation을 필요로 하며, 따라서 필요한 연산의 횟수는 $\frac{5}{2}\ pn^2$ flops이다.

// Let $R_1 \equiv R$;
1. for $k = 1, 2, \cdots, p$
   1.1 Solve triangular system $R_k^T q_k = z_k$;
   1.2 Compute $\rho_k = \sqrt{1 - q_k^T q_k}$;
   1.3 Determine an orthogonal Givens rotation G_i, $(i = n, n-1, \cdots, 1)$ of order $n+1$ so that
   $$G_1 \cdots G_{n-1} G_n \begin{pmatrix} q_k \\ \rho_k \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix};$$
   1.4 Compute $G_1 \cdots G_{n-1} G_n \begin{pmatrix} R_k \\ 0 \end{pmatrix} = \begin{pmatrix} R_{k+1} \\ z_k^T \end{pmatrix}$;

(그림 1) GD 기법
(Fig. 1) GD algorithm

2.2 가상 직교변환에 의한 Downdating

Golub [3]이 제안한 가상 직교변환(pseudo-orthogonalization)은 $S = diag(I_n, -1)$에 대하여 $H^T SH = S$를 만족하는 행렬 H이며, 가상 직교변환 H를 이용하면 간단히 rank 1 downdating을 수행할 수 있다. 특히 2×2 크기의 가상 직교변환

$$H = \begin{pmatrix} \cosh\theta & -\sinh\theta \\ -\sinh\theta & \cosh\theta \end{pmatrix}$$

를 hyperbolic rotation이라고 하며, hyperbolic rotation을 이용한 downdating 기법을 HD(Hyperbolic Downdating) 기법이라고 부른다[1, 5, 7].

HD 기법을 이용한 downdating 알고리즘은 (그림 2)와 같다.

(그림 2)에서 $H^T SH = S = diag(I_n, -1)$이므로 R_{k+1}는 z_k^T로 인하여 R_k을 downdating한 결과임을 알 수

// Let $R_1 \equiv R$;
1. for $k = 1, 2, \cdots, p$
   1.1 Construct and apply a sequence of hyperbolic plane rotation matrices H_i, $(i = n, n-1, \cdots, 1)$ of order $n+1$ so that
   $$H_n \cdots H_2 H_1 \begin{pmatrix} R_k \\ z_k^T \end{pmatrix} = \begin{pmatrix} R_{k+1} \\ 0 \end{pmatrix},$$
   where H_i zeros the i-th element of the current modified vector z_k^T, modifying ith diagonal element of R_k;

(그림 2) HD기법
(Fig. 2) HD algorithm

있다. (그림 2)의 downdating이 완료되면 다행관측행렬 Z^T에 의하여 갱신된 Cholesky factor $\tilde{R} \equiv R_{p+1}$이다.

HD기법은 (그림 2)에서 알 수 있듯이 GD 기법과는 달리 삼각행렬의 해를 구하는 과정이 불필요하므로 하나의 관측행을 downdating할 때마다 GD 기법에 비하여 $\frac{1}{2}n^2$ flops의 시간복잡도가 줄어든다. 따라서 HD 기법이 요구하는 단위 시간당 연산의 시간복잡도는 $2pn^2$ flops이며, GD 기법에 비해 단위 시간당 필요한 연산의 시간복잡도가 작음을 알 수 있다[3, 5, 7]. 그러나 HD 기법은 GD 기법에 비하여 계산상의 정확도(numerical stability)가 다소 떨어지므로 정확도를 높이기 위한 노력이 강구되어야 한다[3, 7, 9].

3. 전처리 과정에 의한 성능향상

3.1 다행관측행렬의 전처리

GD와 HD는 하나의 관측행에 대한 downdating시 항상 n회의 Givens rotation 또는 hyperbolic rotation이 필요하다. 따라서 rotation의 횟수를 줄일 수 있다면 전체적으로 알고리즘의 성능이 개선될 수 있다. 만일 관측치 벡터 z_k^T의 첫 번째부터 v개의 원소가 0이라고 하면,

$$z_k^T = (0 \cdots 0 \times \cdots \times) \equiv (0_v \; x_{n-v}). \tag{9}$$

Preprocess $((Z^T \; \sigma))$
// QR-factorization of Z^T producing R_Z
// Let $(Z^{T(0)} \; \sigma^{(0)}) = (Z^T \; \sigma)$.
1. for $j = 1, 2, \cdots, \min(p, n)$
   1.1. Determine a Householder matrix $Q_Z^{(j)}$, such that $Q_Z^{(j)} Z^{T(j-1)}$ zeros elements below the diagonal of the jth column of $Z^{T(j-1)}$, while $Q_Z^{T(j)} Q_Z^{(j)} = I$;
   1.2. Compute $Q_Z^{(j)}(Z^{T(j-1)} \; \sigma^{(j-1)})$ producing $(Z^{T(j)} \; \sigma^{(j)})$;

(그림 3) Householder reflection을 이용한 전처리 알고리즘
(Fig. 3) Preprocessing based on Householder reflections

그런데 z_k^T를 downdating하는 경우에 영벡터(zero vector) 0_v에 대해서는 rotation이 불필요하므로 rank 1 downdating을 수행하는 데 필요한 연산의 횟수가 그만큼 줄어든다.

예를 들어 GD의 경우, (그림 1)의 단계 1.1에서

$$R_k^T q_k = z_k = \begin{pmatrix} 0_v^T \\ x_{n-v}^T \end{pmatrix} \tag{10}$$

이므로 삼각연립방정식의 해

$$q_k = \begin{pmatrix} 0_v^T \\ q_k^x \end{pmatrix}$$

이다. 여기서 q_k^x는 z_k^T의 0이 아닌 원소로 구성된, 벡터 x_{n-v}^T와 대응하는 원소로 구성된 벡터이다. 따라서 (그림1)의 단계 1.3에서 q_k를 영벡터로 만들기 위한 직교변환 U_k는 식(11)을 만족한다. 즉,

$$U_k \begin{pmatrix} q_k \\ \rho_k \end{pmatrix} = U_k \begin{pmatrix} 0_v^T \\ q_k^x \\ \rho_k \end{pmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

$$U_k = G_{v+1} \, G_{v+2} \cdots G_n. \tag{11}$$

따라서 직교변환 U_k을 계산하기 위해서는 $n-v$회의 Givens rotation이 필요하다.

마찬가지로 HD의 경우에도, $\begin{pmatrix} R_k \\ z_k^T \end{pmatrix} \equiv \begin{pmatrix} R_k^v & B_k \\ O & R_k^{n-v} \\ 0_v & x_{n-v} \end{pmatrix}$에서 z_k^T의 모든 원소를 0으로 만드는 hyperbolic rotation은 $\begin{pmatrix} R_k^{n-v} \\ x_{n-v} \end{pmatrix}$에 대해서만 수행하면 되므로 $n-v$회의 hyperbolic rotation이 필요하다. 따라서 z_k^T의 첫 번째부터 몇 개의 원소가 0이면, 그만큼 rotation의 횟수가 줄어들게 된다.

z_k^T의 원소를 0으로 만들기 위한 방법으로 식(5)와 같이 다행관측행렬 Z^T의 partial Cholesky factor R_Z를 계산하고, 식(6)에 의거하여 R_Z의 각 행에 대한 rank 1 downdating을 수행하면 전체적으로 downdating에 필요한 Givens rotation 또는 hyperbolic rotation의 수를 줄일 수 있다.

Z^T의 Cholesky factor R_Z를 계산하는 과정에서 Z^T가 $p \times n$ 행렬이므로 $p \geq n$인 경우에는 Z^T의 Cholesky factor R_Z는 완전한 상삼각행렬을 이루나, $p < n$인 경우에는 $p \times n$ 행렬인 $R_Z = (R_{Z_1} \; \tilde{Z}_2^T)$의 형태가

된다. 여기서 R_{Z_1}는 $Z^T \equiv (Z_1^T\ Z_2^T)$인 $p \times p$ 행렬 Z_1^T의 Cholesky factor이며,

$$\tilde{Z}_2^T = (Q_{Z_1})^T Z_2^T,\ \ Z_1^T = Q_{Z_1} R_{Z_1},\ (Q_{Z_1})^T Q_{Z_1} = I.$$

따라서 $p < n$인 경우에 R_Z는 완전한 상삼각행렬을 이루지 못하므로 본 논문에서는 이를 partial Cholesky factor라고 정의한다.

(그림 3)은 Z^T를 Householder reflection[13]을 이용하여 partial Cholesky factor R_Z를 계산하는 알고리즘이다. 이 알고리즘은 j값이 증가함에 따라 행렬 $(Z^T\ \sigma)$를 새로운 값으로 갱신하며, 계산이 종료되면 그 결과 $Z^{T(v)}\ \sigma^{(v)}) = (R_Z\ \hat{\sigma})$가 되돌려진다. 여기서 $v = \min(p, n)$이다.

(그림 3)에서 함수 *Preprocess*가 필요로 하는 연산의 시간복잡도는 Golub[12] 및 Kim[13]으로부터

$$\begin{cases} pn^2 - \dfrac{1}{3}n^3 & p \ge n \\ p^2 n - \dfrac{1}{3}p^3 & p < n \end{cases}$$

3.2 PGD (Preprocessed GD)

(그림 3)의 Preprocess를 수행하면 다행관측행렬 Z^T의 partial Cholesky factor R_Z가 계산된다. 여기서 R_Z의 k번째 행을 ψ_k라고 하면,

$$\psi_k \equiv (0_{k-1}\ \ x_{n-k}).$$

그런데 ψ_k는 첫 번째 원소부터 $k-1$ 개의 원소가 0이므로 rank 1 downdating을 수행할 때(식 (10) 참조)

$$R_k^T q_k = \psi_k = \begin{pmatrix} 0_{k-1}^T \\ x_{n-k+1}^T \end{pmatrix}.$$

따라서

$$q_k = \begin{pmatrix} 0_{k-1}^T \\ q_k^x \end{pmatrix}.$$

여기서 q_k^x는 z_k^T의 0이 아닌 원소로 구성된 벡터 x_{n-k+1}^T와 대응하는 원소로 구성된 벡터이다. 또한, q_k를 영벡터로 만들기 위한 Givens rotation은 다음 식을 만족한다(식 (11)참조). 즉,

$$U_k \begin{pmatrix} q_k \\ \rho_k \end{pmatrix} = U_k \begin{pmatrix} 0_{k-1}^T \\ q_k^x \\ \rho_k \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \qquad U_k = G_k G_{k+1} \cdots G_n.$$

이상의 과정을 $k = 1, 2, \cdots, p$에 대하여 순차적으로 수행하여 R_Z를 downdating하는 PGD 알고리즘은 (그림 4)와 같다.

1. *Preprocess* $(Z^T\ \sigma)$;

   // Let $R_Z = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_p \end{pmatrix}$ and $R_1 = R$;

2. for $k = 1, 2, \cdots, \min(p, n)$

   2.1 Solve triangular system $R_k^T q_x^k = \psi_k$;

   2.2 Compute $\rho_k = \sqrt{1 - (q_x^k)^T q_x^k}$;

   2.3 Determine an orthogonal Givens rotation G_i, $(i = n, n-1, \cdots, k)$ of order $n+1$ so that

   $$G_k \cdots G_{n-1} G_n \begin{pmatrix} q_x^k \\ \rho_k \end{pmatrix} = \begin{pmatrix} 0_{n-v}^T \\ 1 \end{pmatrix}$$

   2.4 Compute $G_k \cdots G_{n-1} G_n \begin{pmatrix} R_k \\ 0 \end{pmatrix} = \begin{pmatrix} R_{k+1} \\ \psi_k \end{pmatrix}$;

(그림 4) PGD 기법
(Fig. 4) PGD algorithm

(그림 4)에서 ψ_k를 rank 1 downdating하는데 필요한 회전변환의 횟수는 $n-k+1$이므로 R_Z를 downdating하기 위하여 필요한 Givens rotation의 횟수는

$$\sum_{k=1}^{p} (n-k+1) = np - \frac{p(p-1)}{2}.$$

3.3 PHD (Preprocessed HD)

R_Z의 k번째 행 ψ_k를 rank 1 downdating하기 위한 hyperbolic rotation을 V_k라고 하면,

$$V_k \begin{pmatrix} R_k \\ \psi_k \end{pmatrix} = \begin{pmatrix} R_{k+1} \\ 0 \end{pmatrix}, \qquad V_k = H_n H_{n-1} \cdots H_k.$$

여기서 H_j는 ψ_k의 j번째 원소를 0으로 만들면서 R_k의 j번째 대각원소를 갱신하는 hyperbolic rotation이다. 즉, ψ_k를 downdating하기 위하여 필요한 hyperbolic rotation의 횟수가 $n-k+1$회이므로 R_Z를 downdating하는 데 필요한 hyperbolic rotation의 횟수는

$$\sum_{k=1}^{p}(n-k+1)=np-\frac{p^2-p}{2}$$

이다. PHD에 의한 downdating을 정리하면 (그림 5)와 같다.

1. *Preprocess* $(Z^T\ \sigma)$;
2. for $k=1, 2, \cdots, \min(p, n)$
   2.1 Construct and apply a sequence of hyperbolic plane rotation matrices H_i, $(i=k, k+1, \cdots, n)$ of order $n+1$ so that $H_n \cdots H_{k+1} H_k \begin{pmatrix} R_k \\ \psi_k \end{pmatrix} = \begin{pmatrix} R_{k+1} \\ 0 \end{pmatrix}$,
   where H_i zeros the i-th element of the current modified vector ψ_k;

(그림 5) PHD 기법
(Fig. 5) PHD algorithm

4. 알고리즘의 비교 및 분석

$n\times n$ 행렬 R과 $p\times n$ 행렬 Z^T에 대해 각각 GD와 HD 알고리즘을 이용하여 downdating하는데 필요한 시간복잡도 및 전처리 과정을 거친 PGD와 PHD 알고리즘의 시간복잡도는 〈표 1〉과 같다. 〈표 1〉에서 전처리 알고리즘의 복잡도를 두 가지로 구분한 이유는 $Z^T=Q_Z R_Z$에서 R_Z를 계산할 때, p의 범위에 따라 상이한 형태의 상삼각행렬이 생성되므로 각각의 경우별로 시간복잡도의 계산이 다르기 때문이다. 즉, $p\ge n$인 경우에는 완전 상삼각행렬 R_Z를 얻을 수 있으나 $p<n$인 경우에는 불완전 상삼각행렬을 얻게 된다. 〈표 1〉의 시간복잡도는 수식의 간소화를 위하여 1차항 및 상수항을 생략한 표현으로 〈표 1〉로부터 관측행렬의 행의 수(p)에 따른 시간복잡도의 크기는

PHD < PGD < HD < GD

〈표 1〉 시간복잡도의 비교
〈Table 1〉 Time complexity comparisons

알고리즘	time complexity	
	w/o preprocessing	with preprocessing
GD	$\frac{5}{2}pn^2$	$pn^2+\frac{5}{6}n^3$ if $p\ge n$ $\frac{5}{2}pn^2+\frac{1}{3}p^3-np^2$ otherwisc
HD	$2pn^2$	$pn^2+\frac{1}{3}n^3$ if $p\ge n$ $2pn^2+\frac{1}{3}p^3-np^2$ otherwise

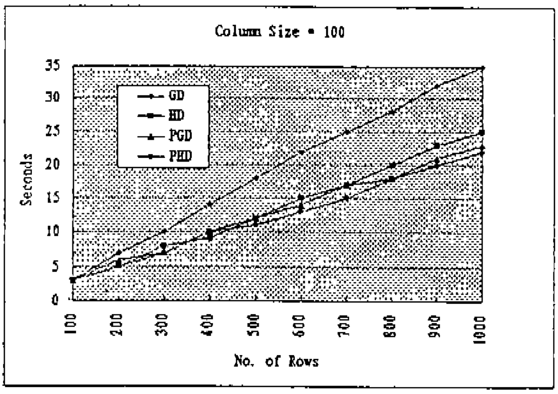
임을 알 수 있다.

이론적인 시간복잡도의 정확성을 규명하기 위하여 Sun SPARC/2 시스템(메모리 16M byte)에서 관측행렬의 크기를 변화시키면서 GD 및 HD 기법과 본 논문에서 제안한 PGD 및 PHD 기법을 이용한 벤치마크 테스트를 수행하였다. 실험 결과는 (그림 6)과 같다. (그림 6) a)는 관측행렬의 열의 크기(n)가 100일 때 행의 크기(p)를 100에서 1,000까지 변화시키면서 계산에 소요되는 시간을 측정한 것이다. 그림에서 알 수 있듯이, PHD에 의한 경우가 PGD, GD 및 HD에 비하여 계산에 소요된 시간이 적으며, 행의 크기가 증가함에 따라 성능 향상이 뚜렷함을 알 수 있다. (그림 6) b)는 열의 크기가 200인 경우에 대한 실험 결과로 열의 크기가 100인 경우와 마찬가지로 PHD기법이 PGD, HD 또는 GD 기법에 비하여 성능이 우수함을 알 수 있다.

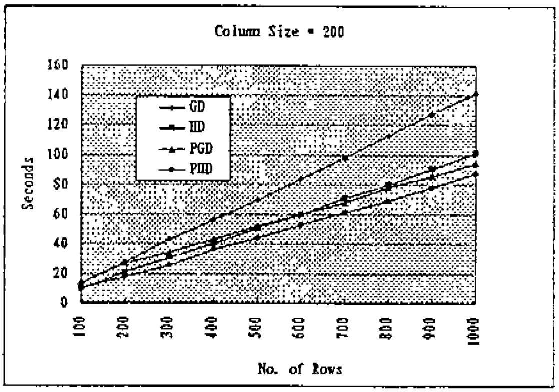
본 논문에서 제안한 기법에서 적용한 전처리 과정이 기존의 알고리즘에 미치는 영향의 정도를 비교하기 위하여 성능향상률(performance improvement rate)을 다음과 같이 정의하였다. 즉, 성능향상률

$$\psi_{PHD}\equiv\frac{t_{HD}}{t_{PHD}},\quad \psi_{PGD}\equiv\frac{t_{GD}}{t_{PGD}}. \tag{12}$$

여기서 t_{PHD}와 t_{HD}는 각각 PHD 및 HD 기법에 의하여 다행관측행렬을 downdating하는데 걸리는 시간이며, t_{PGD}와 t_{GD}는 각각 같은 크기의 다행관측행렬을

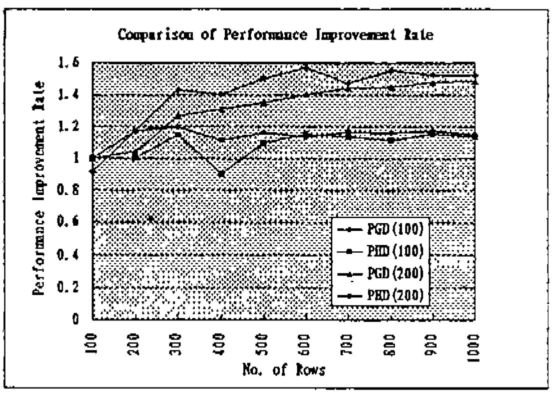


(a) $n=100$



(b) $n=200$

(그림 6) 다행관측행렬의 downdating에 소요된 계산 시간.
(Fig. 6) Computation times for downdating multiple rows



(그림 7) 성능향상률 비교.
(Fig 7) Performance improvement rates

PGD 및 GD 기법을 이용하여 downdating하는데 걸리는 시간이다.

(그림 7)은 식(12)를 토대로 ϕ_{PHD}와 ϕ_{PGD}를 계산하여 그래프로 나타낸 것이다. 열의 크기가 100이거나 200인 경우에 관계없이 항상 $\phi_{PGD} \geq \phi_{PHD}$ 임을 알 수 있으며, 따라서 다행관측행렬의 downdating에서 전처리 과정의 효과는 GD 기법의 경우가 HD 기법의 경우보다 큰 것을 알 수 있다.

5. 결 론

본 논문에서는 신호처리 시스템에 적용할 수 있는 관측치가 연속적으로 입력되는 시스템에서 최소자승해를 구하는 downdating 알고리즘을 연구하였다. 기존의 downdating 방법으로는 Givens rotation을 이용한 GD 기법과 hyperbolic rotation을 이용한 HD 기법이 있으며, 각각을 구현하는 데 필요한 실수 연산의 횟수는 $\frac{5}{2}pn^2$ 및 $2pn^2$이다. 이에 비하여 본 논문에서 제안한 PGD 및 PHD 알고리즘은 관측치 행렬 Z^T를 Householder reflection을 이용하여 QR-분할하는 전처리 과정을 수행하고, 그 결과로 계산된 Z^T의 partial Cholesky factor의 각 행을 기존의 GD 또는 HD 기법을 적용하여 rank 1 downdating하는 두 가지 단계로 구성된다. 따라서 PGD 기법이나 PHD 기법은 기존의 GD 기법이나 HD 기법에 비하여 전처리 과정을 수행하여야 하는 단점이 있으나, rank 1 downdating에 소요되는 계산 시간이 단축되므로 전체적으로 시간복잡도가 개선된다.

연구 결과, PGD 및 PHD 알고리즘의 시간복잡도는 $p \geq n$인 경우 각각 $pn^2+\frac{5}{6}n^3$ 및 $pn^2+\frac{1}{3}n^3$이므로 전처리 과정을 수행하지 않는 기존의 GD 및 HD 기법의 시간복잡도에 비하여 개선되었음을 알 수 있다. 또한, Sun SPARC/2 시스템에서 여러 가지 크기의 관측행렬을 대상으로 하는 실험을 통하여 계산에 소요되는 시간을 측정한 결과도 이론적인 결과와 일치하는 것을 확인하였다. 실험 결과, PHD 기법을 이용한 다행관측행렬의 downdating이 PGD 기법을 이용한 경우에 비하여 우수함을 알 수 있었으며, 전처리 과정이 알고리즘에 미치는 영향은 PGD기법이 PHD 기법에 비하여 큰 것을 알 수 있었다.

참 고 문 헌

[1] S. T. Alexander, 'Adaptive Signal Processing,' Springer-Verlag, New York, 1986.

[2] C. -H. Lee, Sukil Kim, "A fast least-squares algorithm for multiple-row downdatings," Trans. KIPS, Vol. 2, No. 1, pp. 55-65, Jan. 1995.

[3] G. H. Golub, 'Matrix decompositions and statistical calculations, in Statistical Computation,' R. C. Milton and J. A. Nelder, eds., Academic Press, 1969, pp. 365-395.

[4] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations," Math. Comput. Vol. 28, pp. 505-535, 1974.

[5] A. W. Bojanczyk, R. P. Brent, P. Van Dooren, and F. R. de Hoog, "A note on downdating the Cholesky factorization," SIAM Jour. Sci. Statist. Comput. Vol. 8, No. 3, pp. 201-221, Mar. 1987.

[6] A. Bjork, H. Park, and L. Elden, "Accurate downdating of least squares solutions," SIAM Jour. Matrix Anal. Appl., Vol. 15, No. 2, pp. 549-568, April 1994.

[7] S. T. Alexander, C. T. Pan, and R. J. Plemmons, "Analysis of a recursive least squares hyperbolic rotations algorithm for signal processing," Linear Algebra Appl., Vol. 98, pp. 3-40, 1988.

[8] C. S. Henkel, M. T. Heath, and R. J. Plemmons, "Cholesky downdating on a hypercube," In Fox. G.(Ed.), Proc. Third Conference on Hypercube Concurrent Comput. Appl., ACM Press, New York, NY., pp. 1592-1598, 1989.

[9] G. W. Stewart, "The effects of rounding error on an algorithm for downdating a Cholesky factorization," Jour. Inst. Math. Appl., Vol. 23, pp. 203-213, 1979.

[10] C. Bishof and C. F. Van Loan, "The WY representation for products of Householder matrices," SIAM Jour. Sci. Statist. Comput. Vol. 8, No. 1, s2-s13, 1987. 1.

[11] M. A. Saunders, "Large-scale linear programming using the Cholesky factorization," Tech. Report CS252, Computer Science Dept., Stanford University, Stanford, CA, 1972.

[12] G. H. Golub and C. F. Van Loan, 'Matrix Computations,' Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[13] Sukil. Kim, D. P. Agrawal and R. J. Plemmons, "Least-squares multiple updating algorithms on a hypercube," Jour. Par. and Dist. Computing Vol. 8, pp. 80-88, 1990.

[14] C. H. Bischof, C-T. Pan and P. Tang, "A Cholesky up-and downdating algorithm for systolic and SIMD architectures," SIAM Jour. Sci. Comput., Vol. 14, No. 3, pp. 670-676, May 1993.

[15] A. A. Anda and H. Park, "Fast plane rotations with dynamic scaling," SIAM Jour. Mat. Anal. Appl., Vol. 15, No. 1, pp. 162-174, Jan. 1994.

[16] Sukil Kim and C. -H. Lee, "Parallel multiple-row downdatings on distributed memory multiprocessor systems," KISS, submitted for publication, Dec. 1995.

김 석 일

1975년 서울대학교 전기공학과 졸업(학사)

1984년 연세대학교 전자계산학(공학석사)

1989년 North Carolina State University 전기 및 컴퓨터공학(공학박사)

1975년~1990년 국방과학연구소 연구원, 선임연구원

1990년~현재 충북대학교 자연과학대학 컴퓨터과학과 조교수

관심분야: 병렬처리 컴퓨터 구조, 병렬처리 알고리즘, Supercomputing, Advanced factory automation 등.

이 충 한

1994년 충북대학교 컴퓨터과학과(학사)

1994년~현재 충북대학교 전자계산학과 석사과정

관심분야: 병렬처리 컴퓨터구조, 병렬처리 알고리즘.

전 중 남

1981년 연세대학교 전자공학과 공학사

1985년 연세대학교 대학원 공학석사

1990년 연세대학교 대학원 공학박사

1986년~1990년 연세대학교 산업기술 연구소 연구원

1990년~현재 충북대학교 컴퓨터과학과 조교수

관심분야: 컴퓨터구조, 하드웨어 시뮬레이션, 병렬처리