

논문 2015-52-2-19

# 대용량 컴류트 타임 임베딩을 위한 연산 속도 개선 방식 제안

( Proposing the Methods for Accelerating Computational Time of  
Large-Scale Commute Time Embedding )

한 희 일\*

( Hee-il Hahn<sup>Ⓢ</sup> )

요 약

컴류트 타임 임베딩을 구현하려면 그래프 라플라시안 행렬의 고유값과 고유벡터를 구하여야 하는데,  $o(n^3)$ 의 계산량이 요구되어 대용량 데이터에는 적용하기 어려운 문제가 있다. 이를 줄이기 위하여 표본화 과정을 통하여 크기가 줄어든 그래프 라플라시안 행렬에서 구한 다음, 원래의 고유값과 고유벡터를 근사화시키는 Nyström 기법을 주로 채택한다. 이 과정에서 많은 오차가 발생하는데, 이를 개선하기 위하여 본 논문에서는 그래프 라플라시안 대신에 가중치 행렬을 표본화하고 이로부터 구한 고유값과 고유벡터를 그래프 라플라시안의 고유값과 고유벡터로 변환하는 기법을 이용하여 대용량 데이터로 구성된 스펙트럴 그래프를 근사적으로 컴류트 타임 임베딩하는 기법을 제안한다. 하지만, 이 방식도 스펙트럼 분해를 계산하여야 하므로 데이터의 크기가 증가하면 적용하기 어려운 문제가 발생한다. 이의 대안으로, 스펙트럼 분해를 계산하지 않고도 데이터 집합의 크기에 영향을 받지 않으면서 컴류트 타임을 근사적으로 계산하는 방식을 구현하고 이들의 특성을 실험적으로 분석한다.

## Abstract

Commute time embedding involves computing the spectral decomposition of the graph Laplacian. It requires the computational burden proportional to  $o(n^3)$ , not suitable for large scale dataset. Many methods have been proposed to accelerate the computational time, which usually employ the Nyström methods to approximate the spectral decomposition of the reduced graph Laplacian. They suffer from the lost of information by dint of sampling process. This paper proposes to reduce the errors by approximating the spectral decomposition of the graph Laplacian using that of the affinity matrix. However, this can not be applied as the data size increases, because it also requires spectral decomposition. Another method called approximate commute time embedding is implemented, which does not require spectral decomposition. The performance of the proposed algorithms is analyzed by computing the commute time on the patch graph.

**Keywords :** Manifold learning, Commute time embedding, Nyström method, Approximate commute time embedding

\* 평생회원, 한국외국어대학교 정보통신공학과  
(Department of Information and Communications  
Engineering, Hankuk University of Foreign  
Studies)

Ⓢ Corresponding Author(E-mail: hihahn@hufs.ac.kr)

※ 이 논문은 2014년도 한국외국어대학교 교내 학술연구  
구지원에 의하여 연구되었음.

접수일자: 2014년11월15일, 수정일자: 2015년01월19일  
게재확정: 2015년01월29일

## I. 서 론

데이터를 비선형 다양체 위의 점으로 간주하여 이를 저 차원 공간으로 임베딩함으로써 기하학적으로 차원을 줄이려는 다양체 학습(manifold learning) 기법이 최근 들어 활발히 연구되고 있다. 이들의 대표적인 예로는

ISOMAP<sup>[12]</sup>, LLE(locally linear embedding)<sup>[9]</sup>, 라플라시안 아이젠맵(Laplacian eigenmap)<sup>[2]</sup>, 컴뮷트 타임 임베딩(commute time embedding: 이하 CTE라 부름)<sup>[7~8]</sup> 등이 있다. 다양체 학습 방식은 기본적으로 입력 데이터 집합  $X$  가 놓여 있는 공간에서 보다 이들의 특징벡터 공간  $Y$  에서 그 구조가 보다 더 명확하도록 하기 위해서 커널 방식을 이용한다. 커널 방식에서는 입력 데이터가 맵  $\phi: X \rightarrow Y$  를 통해서 특징벡터 공간으로 매핑될 때 커널 행렬이 정의된다. 이 커널 행렬  $G \in \mathbb{R}^{n \times n}$  은 그람(Gram) 행렬이라고도 부르는데,  $G_{ij} = \kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$  와 같이 각 원소는 특징 벡터 간의 내적으로 표현될 수 있다. 이 과정에서 그람 행렬의 고유값과 고유벡터를 구하여야 하는데, 데이터 개수를  $n$  이라 할 때  $O(n^3)$  의 계산량이 요구된다.  $n$  이 충분히 크면 현재의 컴퓨팅 기술로는 구현이 매우 어려우므로 이를 해결하기 위하여 랜덤 샘플링이나 저 랭크 행렬 근사화 등의 기법을 이용한다. 이의 대표적인 방식으로는 Williams<sup>[13]</sup>가 커널 기반 패턴분류기의 속도 향상을 목적으로 그람 행렬의 근사화를 위해 적용한 Nyström 근사화 방식이 있다.

이와 같이 그람 행렬에 대한 근사화는 많은 연구가 진행되었으나, 라플라시안 아이젠맵이나 컴뮷트 타임 임베딩 등에서도 같이 그람 행렬 대신 라플라시안이나 정규 그래프 라플라시안(normalized graph Laplacian) 등을 근사화시켜야 하는 경우에는 계산 과정에서 많은 사황을 고려하여야 한다. 예를 들어, Choromanska<sup>[3]</sup>는 SPSD(symmetric, positive semidefinite) 인 정규 그래프 라플라시안 행렬에서 고유값과 고유 벡터를 구하는 알고리즘을 제안하고 있는데, 그래프 라플라시안 행렬에서 랜덤으로 열을 표본화한 다음 정규화하는 과정에서 구한 부 행렬(submatrix)은 더 이상 SPSD하지 않으므로 근사화 오차가 크게 발생하는 단점이 있다. 따라서 본 논문에서는 이러한 문제를 해결하기 위하여 그래프 라플라시안 행렬 대신에 그래프의 가중치 행렬을 표본화하고 이로부터 구한 고유값과 고유벡터를 그래프 라플라시안의 고유값과 고유벡터로 변환하는 방법을 이용하여 대용량 데이터로 구성된 스펙트럴 그래프를 근사적으로 컴뮷트 타임 임베딩하는 기법을 제안한다. 또 다른 방법으로는 그래프 라플라시안 행렬의 고유값과 고유벡터를 직접 추정하는 대신에, 이들로부터 유도되는 컴뮷트 타임을 근사적으로 추정하는 방식

이 가능하다. 컴뮷트 타임은 기본적으로 라플라시안의 고유값과 고유벡터를 이용하여 계산할 수 있으나, Johnson-Lindenstrauss 정리<sup>[11]</sup>와 Spielman-Teng 알고리즘<sup>[11]</sup> 등을 이용하면 엄청난 계산량이 요구되는 고유값과 고유벡터를 구하지 않고도 비교적 정확하게 컴뮷트 타임을 근사적으로 구할 수 있다. 본 논문에서는 제안 알고리즘들의 동작을 실험적으로 확인하기 위하여 파형 신호에서 서로 겹치도록 추출한 패치를 그래프로 구성한다.

본 논문의 구성은 다음과 같다. II장에서는 컴뮷트 타임 임베딩을 리뷰하고, Nyström 기법을 이용하여 정규 라플라시안의 고유값과 고유벡터의 근사값을 구하는 알고리즘을 III장에서 설명한다. IV장에서는 그래프 라플라시안의 고유값과 고유벡터를 구하지 않고 컴뮷트 타임을 근사적으로 구하는 알고리즘을 설명한다. 이 패치 그래프의 정규 그래프 라플라시안에서 구한 고유값과 고유벡터를 이용하여 구현한 컴뮷트 타임 임베딩과, 본 논문에서 제안한 두 방식을 각각 구현하고 이들의 특성과 성능에 대한 비교 분석을 V장에 제시한다. 마지막으로 VI장에서는 결론을 맺고 향후 연구 진행방향에 대하여 논의한다.

## II. 컴뮷트 타임 임베딩 리뷰

### 1. 그래프 라플라시안

그래프는 기본적으로 무향(undirected)이고 가중 행렬  $W$ 는 대칭 즉,  $w(u, v) = w(v, u)$  이라고 가정한다. 그래프 라플라시안 행렬  $L$  은  $D - W$  로 정의되는데, 여기서  $D$  는 각 원소가  $d_u = \sum_v w(u, v)$  인 대각행렬이다.  $L$  은 SPSD 이고 그 고유값은 항상 음이 아닌 값을 갖는다. 또한, 그래프가 완전 연결일 때, 0의 고유값이 존재하고 이에 해당되는 고유 벡터는  $\mathbf{1} = [1, \dots, 1]^T$ 이다. 일반적으로 행렬  $L$ 을 다음과 같은 두 가지 방법으로 정규화하기도 한다.

$$\begin{aligned} L_{sym} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \\ L_{rw} &= D^{-1} L = I - D^{-1} W \end{aligned} \quad (1)$$

$L_{rw}$ 의 고유값과 고유벡터가 각각  $\lambda, v$  즉,  $L_{rw} v = \lambda v$  이면  $L_{sym}$ 의 고유값은  $L_{rw}$ 의 것과 동일한  $\lambda$  이고 해당 고유벡

터는  $D^{1/2}v$  즉,  $L_{sym}D^{1/2}v = \lambda D^{1/2}v$ 이 성립한다. 또한  $Lv = \lambda Dv$ 이면  $L_{rw}$ 의 고유값과 고유벡터는 각각  $\lambda, v$ 이다. 이러한 특성을 이용하면 정규 또는 비정규 라플라시안 행렬을 이용하여 그래프를 다양하게 해석할 수 있는 장점이 있다<sup>[4]</sup>.

## 2. 컴퓨트 타임 임베딩

CTE는 데이터 집합  $x_1, \dots, x_n \in \mathbb{R}^p$ 가 주어질 때, 이들의 정보손실을 최소한으로 줄이면서 보다 낮은 차원의 점  $y_1, \dots, y_n \in \mathbb{R}^q$ 으로 변환시키기 위하여, 스펙트럴 그래프 이론을 이용하여 적절한 맵을 구한다<sup>[4]</sup>. 두 노드  $x_i$ 와  $x_j$  간의 컴퓨트 타임  $c(x_i, x_j)$ 는 랜덤 워크가  $x_i$ 에서  $x_j$ 로 이동한 다음 다시  $x_i$ 로 되돌아오는 데 소요되는 평균시간으로 정의된다. 그래프에서 최단 거리(즉 지거리)와는 달리, 두 노드 간의 컴퓨트 타임은 이들을 연결하는 경로가 많을수록 감소한다. 두 노드  $x_i$ 와  $x_j$  간의  $c(x_i, x_j)$ 는 다음과 같이 구할 수 있다<sup>[8]</sup>.

$$\begin{aligned} c(x_i, x_j) &= V_G(l_{ii}^\dagger - 2l_{ij}^\dagger + l_{jj}^\dagger) \\ &= V_G(e_i - e_j)^T L^\dagger (e_i - e_j) \end{aligned} \quad (2)$$

여기서  $e_i = [0, \dots, 0, 1, 0, \dots, 0]^T$ 이고  $L^\dagger = (l_{ij}^\dagger)_{i,j=1,\dots,n}$ 는  $L$ 의 Moore-Penrose 역행렬이다. 즉,  $L = UAU^T$ 로 스펙트럼 분해(spectral decomposition) 된다면  $L^\dagger = UA^\dagger U^T$ 이다. 여기서,  $U = [u_1, \dots, u_n]$ 의 각 열 벡터  $u_i$ 는  $L$ 의 고유벡터이고  $\lambda_i$ 는  $u_i$ 에 해당되는 고유값  $\lambda_i$ 을 대각 원소로 하는 대각행렬이며,  $A^\dagger$ 는  $A$ 의 Moore-Penrose 역으로 다음과 같이 정의된다.

$$\lambda_i^\dagger = \text{diag} \begin{cases} \frac{1}{\lambda_i} & \lambda_i \neq 0 \\ 0 & \lambda_i = 0 \end{cases} \quad (3)$$

식 (2)의  $c(x_i, x_j)$ 는 다음과 같이 나타낼 수 있는데,

$$\begin{aligned} c(x_i, x_j) &= V_G(e_i - e_j)^T U A^\dagger U^T (e_i - e_j) \\ &= (\psi_i - \psi_j)^T (\psi_i - \psi_j) \end{aligned} \quad (4)$$

여기서,  $d_i = \sum_{j=1}^n w(i, j)$ 이고  $V_G = \sum d_i$ 이며,

$$\psi_i = \sqrt{V_G} \left( \frac{u_{i,2}}{\sqrt{\lambda_2}}, \dots, \frac{u_{i,n}}{\sqrt{\lambda_n}} \right)^T \text{인 데, 이의 행렬 표현}$$

은  $\psi_i = \sqrt{V_G} (\Lambda^\dagger)^{1/2} U^T e_i$ 이다.  $L$  대신  $L_{sym}$ 을 이용하면

$$\psi_i = \sqrt{V_G} \left( \frac{v_{i,2}}{\sqrt{\lambda_2 d_i}}, \dots, \frac{v_{i,n}}{\sqrt{\lambda_n d_i}} \right) \quad (5)$$

으로 변경하면 된다. 여기서,  $\hat{\lambda}_i = \frac{\lambda_i}{\sqrt{d_i}}$ 이고

$v_{i,j} = \frac{u_{i,j}}{\sqrt{d_i}}$ 이다. 이를 행렬로 표현하면 각각

$\hat{\Lambda} = \Lambda D^{-1/2}$ ,  $V = D^{-1/2} U$ 이다. 위 식 (4)를 관찰하면  $\sqrt{c(x_i, x_j)}$ 는  $\mathbb{R}^{n-1}$  상에서 두 벡터 간의 거리로 해석될 수 있다. 즉,  $x_i$ 를 식 (5)와 같이 임베딩하면  $\sqrt{c(x_i, x_j)}$ 는 임베딩 공간의 두 노드  $\psi_i$ 와  $\psi_j$  간의 유클리드 거리로 간주될 수 있는데, 이를 컴퓨트 타임 거리라고 부른다. 임베딩 공간의 차원은 노드 수에 따라 증가하므로 어느 정도의 오차를 허용하는 범위 내에서 차원을  $q$ 로 줄이면 다음과 같이 CTE를 정의할 수 있다<sup>[6-8]</sup>.

$$\phi_i = \sqrt{V_G} \left( \frac{v_{i,2}}{\sqrt{\lambda_2 d_i}}, \dots, \frac{v_{i,q+1}}{\sqrt{\lambda_{q+1} d_i}} \right) \quad (6)$$

여기서 고유값  $\hat{\lambda}_k$ 는 다음과 같이 정렬되어 있다고 가정한다.

$$0 = \hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \dots \leq \hat{\lambda}_n < 2 \quad (7)$$

## III. Nyström 방식을 이용한 컴퓨트 타임 임베딩

### 1. Nyström 기법 리뷰

SPSD인  $n \times n$  그람 행렬  $G$ 가 주어지면,  $G = UAU^T$ 으로 나타낼 수 있다. 여기서  $\Lambda$ 는  $G$ 의 고유값이 원소인 대각 행렬이고  $U$ 는 각 열이 고유값에 대응하는 고유벡터로 구성된 행렬이다.  $G$ 의  $n$ 개 열 벡터 중에서  $c$ 개를 균일한 확률로 랜덤하게 추출하여  $n \times c$  행렬  $C$ 를 구성하고, 그  $c$ 개의 열 벡터와 그에 해당하는  $c$ 개의 행 벡터가 교차하여 생성된  $c \times c$  행렬  $\hat{G}$ 를 구한다.  $G$ 가 SPSPD이므로  $\hat{G}$ 도 SPSPD임을 알 수 있다. 편의 상,  $G$ 의 행과 열을 재배열시키면 다음과 같이 나타낼 수 있다.

$$G = \begin{bmatrix} \hat{G}A^T \\ A \ B \end{bmatrix}, \quad C = \begin{bmatrix} \hat{G} \\ A \end{bmatrix} \quad (8)$$

Nyström 방식<sup>[13]</sup>에 의하면  $G$ 는  $\tilde{G} = C\hat{G}_k^\dagger C^T$ 으로 근사화시킬 수 있다. 여기서  $\hat{G}_k$ 는  $\hat{G}$ 의 랭크(rank)가  $1 \leq k \leq c$ 가 되도록 조정된 행렬이고  $\hat{G}_k^\dagger$ 는  $\hat{G}_k$ 의 Moore Penrose 역 행렬이다.  $c$ 가 증가함에 따라  $\tilde{G}$ 는  $G$ 에 수렴하고  $\tilde{G} = \tilde{U}\tilde{\Lambda}\tilde{U}^T$ 으로 분해될 수 있다. 여기서  $\tilde{U}$ 과  $\tilde{\Lambda}$ 는 다음과 같이 주어진다.

$$\tilde{U} = \sqrt{\frac{c}{n}} CU_{\hat{G}}A_{\hat{G}}^\dagger, \quad \tilde{\Lambda} = \frac{n}{c}A_{\hat{G}} \quad (9)$$

위 식에서  $\hat{G} = U_G A_G U_G^\dagger$ 이다. 즉,  $G$ 의 고유값  $\Lambda$ 과 고유벡터  $U$ 는  $\tilde{\Lambda}$ 과  $\tilde{U}$ 으로 각각 근사화시킬 수 있다.  $G$ 를 랭크  $k$ 로 근사화시키기 위해서는  $\hat{G}_k$ 의 고유값과 고유벡터를 구하고 행렬  $C$ 와 곱셈하는데  $\mathcal{O}(c^3 + kcn)$ 의 계산량이 요구되므로  $c \ll n$ 일 때에는 계산량을 크게 줄일 수 있는 특징이 있다.

2. 정규 라플라시안의 고유값과 고유벡터의 근사화  
주어진 가중치 행렬  $W$ 에서 균일 확률로  $c$ 개의 열을 랜덤 추출하여 행렬  $C \in \mathbb{R}^{n \times c}$ 를 생성하고 동일한 인덱스에 해당되는 행을 추출하여  $A \in \mathbb{R}^{c \times c}$ 를, 나머지 행을 모아서  $B \in \mathbb{R}^{(n-c) \times c}$ 를 각각 구한다.  $W$ 의 행과 열을 재배열시키면 다음과 같이 나타낼 수 있다.

$$W = \begin{bmatrix} A & B^T \\ B & H \end{bmatrix}, \quad C = \begin{bmatrix} A \\ B \end{bmatrix} \quad (10)$$

$W$ 는 대칭행렬이므로  $A$ 도 대칭이다. 따라서 3.1절에서 설명한 바와 같이  $A$ 에서 정도 행렬(degree matrix)  $D_A$ 를 구하여 아래 알고리즘의 3 단계에 제시한 바와 같이  $A$ 를 정규화함으로써 행렬  $\hat{A}$ 를 구성한다.  $B$ 는 비대칭 행렬이므로 정규화하기 위하여 본 논문에서는  $B$ 의 행과 열을 이용하여 정도 행렬  $D_B$ 와  $\Delta_B$ 를 별도로 구한 다음, 4 단계에 설명한 대로  $B$ 를 정규화시킨다. 그런데  $B$ 의 행과 열의 수가 서로 다르기 때문에  $\Delta_B$ 에  $\frac{c}{n-c}$ 를 곱하여  $D_B$ 와 스케일이 근사하도록 조정하여야 한다. 마지막으로  $W$ 의 고유값  $\Lambda_W$ 과 고유벡터  $U_W$ 에 대한 근사값을 구하고,  $L_{sym}$ 의 고유값

$\Lambda = 1 - \Lambda_W$ 이고  $W$ 와  $L_{sym}$ 은 동일한 고유벡터를 갖는다는 사실을 이용하여 7 단계에 나타낸 바와 같이  $\hat{A}$ 의 고유값과 고유벡터로부터  $L_{sym}$ 의 고유값과 고유벡터에 대한 근사값을 각각 구한다.

Choromanska<sup>[3]</sup>가 제안한 방식은 행렬  $C$ 에서 직접 정도 행렬  $D$ 와  $\Delta$ 를 구하여  $\tilde{L}_{sym} \leftarrow \hat{I} - \sqrt{\frac{c}{n}} D^{-1/2} C \Delta^{-1/2}$ 를 계산한다. 여기서  $\hat{I} \leftarrow \mathcal{I}(:, \Omega)$ 이고  $D \in \mathbb{R}^{n \times n}$ 와  $\Delta \in \mathbb{R}^{c \times c}$ 는 다음과 같이 정의된다.

$$D(i, i) = \sum_{j=1}^c \mathcal{O}(i, j), \quad \Delta(i, i) = \sum_{j=1}^n \mathcal{O}(j, i) \quad (11)$$

그런 다음,  $\hat{L}_{sym} \leftarrow \tilde{L}_{sym}(\Omega, :)$ 의 고유값과 고유벡터를 이용하여 근사값을 구한다. 그런데,  $L_{sym}$ 은 SPSP인 데 반하여  $\hat{L}_{sym}$ 은 그렇지 않기 때문에 이 방식으로 구한 고유값과 고유벡터는 실제 값에서 크게 벗어남과 동시에  $L_{sym}$ 의 최소 고유값은 0임을 만족시키지 못한다. Choromanska 방식이  $\hat{L}_{sym}$ 에서 고유값과 고유벡터를 구하는데 반하여, 본 제안 방식은 가중치 행렬  $W$ 의 샘플 버전인  $A$ 에서 구하고,  $L_{sym}$ 의 고유값  $\Lambda = 1 - \Lambda_W$ 이고  $W$ 와  $L_{sym}$ 은 동일한 고유벡터를 갖는다는 사실을 이용함으로써 보다 정확하게 추정할 수 있는 장점이 있다.  $L_{sym}$ 의 고유값과 고유벡터가 근사적으로 구해지면 식 (6)을 이용하여 컴뮷트 타임 임베딩할 수 있다. 이 때, 임베딩 공간은  $\mathbb{R}^{(c-1) \times (c-1)}$ 이다.

---

#### 정규 라플라시안 $L_{sym}$ 의 Nyström 근사화 알고리즘

---

입력 : 가중치 행렬  $W \in \mathbb{R}^{n \times n}$ , 표분화할 열 벡터 수  $c$   
출력 :  $L_{sym}$ 의 근사 고유값과 고유벡터  $\tilde{\Lambda}, \tilde{U} \in \mathbb{R}^{n \times c}$

1.  $C \leftarrow W(:, \Omega)$ , 여기서  $\Omega$ 는 균일 확률로 표분화한  $c$ 개의 인덱스 집합
  2.  $A \leftarrow C(\Omega, :), B \leftarrow C(\Omega^c, :)$ ,  $\Omega^c$ 는  $\Omega$ 의 여집합
  3.  $\hat{A} \leftarrow D_A^{-1/2} A D_A^{-1/2}$ , 대각 행렬  $D_A(i, i) = \sum_{j=1}^c A(i, j)$
  4.  $\hat{B} \leftarrow D_B^{-1/2} B \Delta_B^{-1/2}$ , 대각 행렬  $D_B(i, i) = \sum_{j=1}^n B(i, j)$ ,
-

$$\Delta_B(i, i) = \frac{c}{n-c} \sum_{j=1}^{n-c} B(j, i)$$

5.  $\hat{A} = U_{\hat{A}} A_{\hat{A}} U_{\hat{A}}^T$  으로 spectral decomposition

6.  $\hat{C} \leftarrow \begin{bmatrix} \hat{A} \\ \hat{B} \end{bmatrix}_{\Omega}$ ,  $\Omega$  의 역순으로 재배열시킨다.

7.  $\tilde{A} = 1 - \frac{n}{c} A_{\hat{A}}$ ,  $\tilde{U} = \sqrt{\frac{c}{n}} \hat{C} U_{\hat{A}} \tilde{A}^\dagger$

#### IV. 컴퓨터 타임 거리의 근사화를 통한 컴퓨터 타임 임베딩

식 (5)에 주어진 바와 같은 컴퓨터 타임 임베딩을 구현하려면  $L_{sym}$ 의 고유값과 고유벡터를 구하는데  $O(n^3)$ 의 계산량이 요구된다.  $n$ 이 2,000 이상의 큰 값을 가질 때에는 엄청난 계산량으로 인하여 구현이 사실상 불가능하다. 하지만, Johnson-Lindenstrauss<sup>[1]</sup>, Spielman이 제안한 방식<sup>[11]</sup>을 이용하여 근사화된 컴퓨터 타임 임베딩(approximate commute time embedding : 이하 ACTE라 부름)을 구하는 것이 가능하다<sup>[10]</sup>. 우선, 그래프에서 에지의 연결성을 나타내는  $m \times n$  행렬  $B$ 를 다음과 같이 정의한다.

$$B(u, v) = \begin{cases} 1 & \text{if } v \text{ is } u\text{'s head} \\ -1 & \text{if } u \text{ is } v\text{'s head} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

여기서  $m$ 은 에지의 수를 나타낸다. 이와 함께  $m \times m$  대각 행렬  $H$ 를 정의하는데,  $H(i, i)$ 는  $i$ 번째 에지의 가중치이다. 그러면, 라플라시안 행렬  $L = B^T H B$ 를 구할 수 있다. 이를 이용하면 두 노드  $x_i$ 와  $x_j$ 간의 컴퓨터 타임  $c(x_i, x_j)$ 을 다음과 같이 유도할 수 있다.

$$\begin{aligned} c(x_i, x_j) &= V_G(e_i - e_j)^T L^\dagger (e_i - e_j) \\ &= V_G(e_i - e_j)^T L^\dagger L L^\dagger (e_i - e_j) \\ &= V_G(e_i - e_j)^T L^\dagger B^T H B L^\dagger (e_i - e_j) \\ &= \left\| \sqrt{V_G} H^{1/2} B L^\dagger (e_i - e_j) \right\|^2 \end{aligned}$$

따라서  $c(x_i, x_j)$ 는  $\sqrt{V_G} H^{1/2} B L^\dagger$ 의 각 열 벡터 간의 거리의 제곱으로 표현됨을 알 수 있다. Johnson-Lindenstrauss 정리에 의하면, 이 벡터들을  $O(\log n)$ 개의 랜덤 벡터가 스캔하는 공간으로 사영시키더라도 거리를 보존한다<sup>[1]</sup>. 즉, 임의의 두 벡터  $v_i, v_j \in \mathbb{R}^d$ 와 상

수  $\epsilon > 0$ 에 대하여  $k \times d$  랜덤 행렬  $Q$ 가 주어지면  $1 - \frac{1}{n}$  이상의 확률로 다음 부등식을 만족시킨다.

$$(1-\epsilon) \|v_i - v_j\|^2 \leq \|Qv_i - Qv_j\|^2 \leq (1+\epsilon) \|v_i - v_j\|^2$$

여기서  $k \propto \log \frac{n}{\epsilon^2}$ 이다<sup>[10]</sup>. 위 식을 컴퓨터 타임에 적용하면 다음과 같은 부등식을 얻을 수 있다<sup>[10]</sup>.

$$(1-\epsilon)c(x_i, x_j) \leq \|Ze_i - Ze_j\|^2 \leq (1+\epsilon)c(x_i, x_j) \quad (13)$$

여기서  $Z = \sqrt{V_G} Q H^{1/2} B L^\dagger$ 이다. 즉,  $c(x_i, x_j)$ 는  $\|Z(e_i - e_j)\|^2$ 으로 근사화시킬 수 있다. 그런데,  $L$ 의 역행렬인  $L^\dagger$ 를 구하는 데에는 엄청난 계산량이 요구되므로 다음과 같은 방법을 이용한다.

우선,  $Y = \sqrt{V_G} Q H^{1/2} B$ 를 계산한 후에,  $Z = YL^\dagger$ 이므로  $ZL = Y$ 을 만족하는  $Z$ 를 구함으로써 컴퓨터 타임의 근사값을 구한다. 이를 위하여 본 논문에서는 Spielman-Teng 알고리즘을 이용하여  $Z$ 의 모든 행에 대하여  $z_i L = y_i$ 의 근을 구한다. 여기서,  $z_i, y_i$ 는 각각  $Z$ 와  $Y$ 의  $i$ 번째 행벡터를 나타낸다. 이와 같이  $Z$ 를 구하면 다음과 같이 근사화된 컴퓨터 타임 임베딩을 구할 수 있다.

$$\chi_i = \sqrt{V_G}(z_{i,1}, z_{i,2}, \dots, z_{i,k}) \quad (14)$$

여기서,  $z_{i,r}$ 는  $Z$ 의  $i$ 번째 열의  $r$ 번째 성분이다.

#### V. 실험 및 토론

본 논문에서는 700개의 샘플로 구성된 정현파 신호에서 25차원 벡터 크기의 패치를 최대한 겹치도록 (maximally overlapped) 676개 추출한 다음, 이를 패치 그래프로 구성하여 실험한다<sup>[6~7]</sup>.

##### 1. Nyström 기법을 이용한 컴퓨터 타임 임베딩

패치 그래프에 식 (6)을 이용하여 컴퓨터 타임 임베딩한 결과를 [그림 1.]에 제시하였다. 이 그림에서는 각 패치에서 그레이던트 벡터의 크기를 각각 구하고 이를 오름 차순으로 정렬하여 중간 값보다 작으면 파란 점으로, 크면 빨간 점으로 나타낸다. 컴퓨터 타임 임베딩에서는 부드러운 곡선 구조를 갖고 있는데 파란 점과 빨

간 점이 각각 클러스터링되어 있는 모습을 볼 수 있다.

[그림 2.]는 위의 패치 그래프에서 구한 정규 라플라시안  $L_{sym}$ 의 고유값과 고유벡터를 Nyström 방식으로

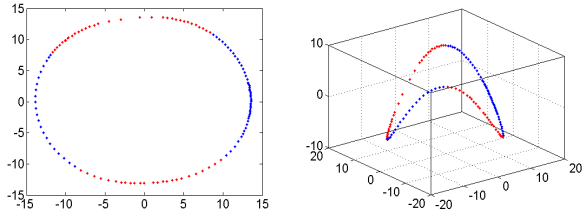


그림 1. 정현파 신호에 적용한 패치 그래프를 이용한 컴्यू트 타임 임베딩

좌: 이차원 임베딩, 우: 삼차원 임베딩( $q=3$ )

Fig. 1. Commute time embedding of a sinusoidal signal from which the patch graph is constructed. Left: Two-dimensional ( $q=2$ ) Embedding, Right : Three-dimensional ( $q=3$ ) embedding.

근사화한 다음, 식 (6)을 이용하여 컴्यू트 타임 임베딩한 결과를 보여주고 있다. 여기서 좌열은 Choromanska의 방식으로 구한 결과를, 우열은 제안 방식으로 구한 결과를 각각 나타내고,  $L_{sym}$ 의 676 개 열 중에서 200, 300, 400, 500, 600개의 열을 균일 샘플링하여 구한 결과를 첫 행부터 순서대로 나타낸다.

이 그림에서 알 수 있듯이, Choromanska의 결과는 샘플링된 패치에 해당되는 고유벡터의 성분이 그렇지 않은 패치에서 보다 스케일이 훨씬 작고, 샘플링된 열의 수에 따라서도 스케일이 크게 변하는 문제가 있다. 그 결과, 두 개의 스케일이 다른 임베딩이 겹쳐 보이는 현상이 나타난다. 하지만, 제안 방식의 결과는 샘플링된 열의 수가 증가함에 따라 컴्यू트 타임 임베딩의 참값 [그림 1.]에 수렴할 뿐만 아니라 스케일도 거의 동일함을 알 수 있다.  $L_{sym}$ 의 고유값은 식 (7)에 주어진 바

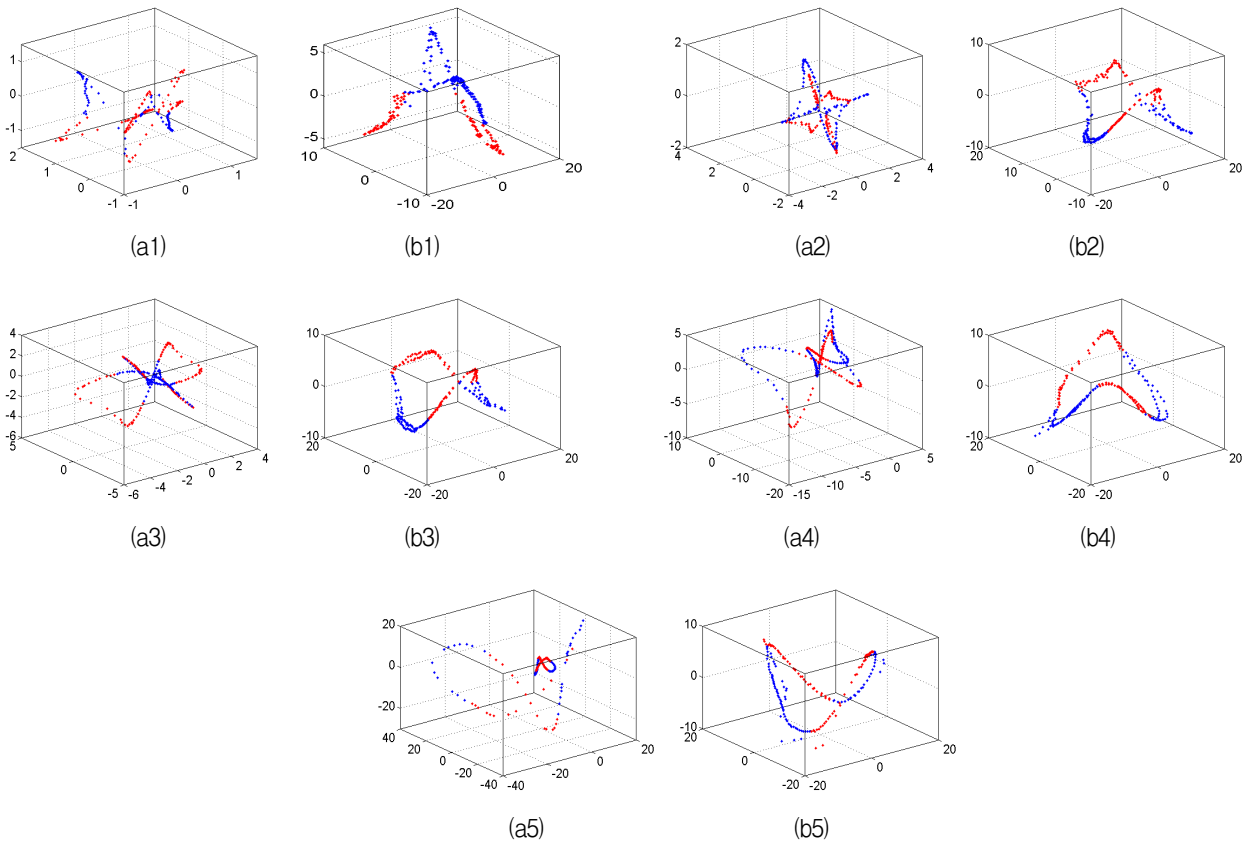


그림 2. Nyström 기법을 이용한 컴्यू트 타임 임베딩

좌열: Choromanska 방식, 우열: 제안 방식, 표본화율 : 1행(29.6%), 2행(44.4%), 3행(59.2%), 4행(73.9%), 5행(88.7%)

Fig. 2. Commute time embedding using Nyström method. Left two columns: Choromanska's method, Right two columns: The proposed method, Sampling rates : The first row(29.6%), second row(44.4%), third row(59.2%), fourth row(73.9%), fifth row(88.7%).

와 같이  $[0, 2)$  범위 내에 존재하게 되는데, 위 패치 그래프에서 구한 고유값을 오름 차순으로 배열한 5 개의 가장 작은 고유값은 [표 1.] 에 제시한 바와 같다.

표 1.  $L_{sym}$  의 가장 작은 5개 고유값 (참값)  
Table 1. The five smallest eigenvalues of the  $L_{sym}$

|     | $\lambda'_1$ | $\lambda'_2$ | $\lambda'_3$ | $\lambda'_4$ | $\lambda'_5$ |
|-----|--------------|--------------|--------------|--------------|--------------|
| 고유값 | 0.0          | 0.010        | 0.012        | 0.040        | 0.045        |

표 2. Choromanska 방식으로 구한, 샘플링 열의 수에 따른 5개의 가장 작은 고유값  
Table 2. The five smallest eigenvalues of the  $L_{sym}$ , obtained using the Choromanska's method.

| 샘플수 | $\lambda'_1$ | $\lambda'_2$ | $\lambda'_3$ | $\lambda'_4$ | $\lambda'_5$ |
|-----|--------------|--------------|--------------|--------------|--------------|
| 200 | 2.304        | 2.331        | 2.340        | 2.359        | 2.425        |
| 300 | 1.198        | 1.219        | 1.238        | 1.255        | 1.274        |
| 400 | 0.637        | 0.650        | 0.711        | 0.729        | 0.745        |
| 500 | 0.339        | 0.346        | 0.350        | 0.379        | 0.410        |
| 600 | 0.119        | 0.131        | 0.136        | 0.166        | 0.175        |

표 3. 제안 방식으로 구한, 샘플링 열의 수에 따른 5 개의 가장 작은 고유값  
Table 3. The five smallest eigenvalues of the  $L_{sym}$ , obtained using the proposed method.

| 샘플수 | $\lambda'_1$ | $\lambda'_2$ | $\lambda'_3$ | $\lambda'_4$ | $\lambda'_5$ |
|-----|--------------|--------------|--------------|--------------|--------------|
| 200 | 0.0          | 0.006        | 0.009        | 0.027        | 0.036        |
| 300 | 0.0          | 0.008        | 0.010        | 0.031        | 0.038        |
| 400 | 0.0          | 0.009        | 0.011        | 0.038        | 0.041        |
| 500 | 0.0          | 0.009        | 0.011        | 0.037        | 0.046        |
| 600 | 0.0          | 0.010        | 0.011        | 0.040        | 0.046        |

표 4. 패치의 표본화 샘플 수에 따른 제안 방식의 컴퓨터 타임 근사화 오차  
Table 4. The approximation errors of the proposed method as a function of the number of sampled patches

|       | 200   | 300   | 400   | 500  | 600 |
|-------|-------|-------|-------|------|-----|
| $e_p$ | 500.8 | 189.1 | 106.9 | 22.9 | 9.7 |

위에서 설명한 바와 같이,  $L_{sym}$  의 가장 작은 고유값은 항상 0 이어야 하고 또한 2 보다 작은 양의 고유값을 가져야 함에도 불구하고 Choromanska 방식은 [표 2.] 에 나타낸 바와 같이 이를 만족시키지 못하며, 샘플링 열의 수가 작을수록 크게 벗어남을 확인할 수 있다.

하지만, 본 논문에서 제안한 방식은 [표 3.]에 제시한 바와 같이 위의 두 조건에 부합할 뿐만 아니라 샘플링 열 수가 증가할수록 실제 고유값에 수렴함을 알 수 있다. 또한, 제안 방식의 컴퓨터 타임 정확도를 확인하기 위하여 다음과 같이 오차를 정의하여 표본화 샘플 수에 따른 평균 제곱 오차의 변화를 [표 4.]에 제시하였다.

$$e_p = \frac{1}{n} \sum_{i=1}^n \|\psi_i - \hat{\psi}_i\|^2 \tag{15}$$

여기서,  $\hat{\psi}_i$  는 근사적으로 구한 컴퓨터 타임 임베딩을 나타낸다.

### 2. 근사 컴퓨터 타임 임베딩

위의 4 절에 설명한 바와 같이, 그래프 라플라시안의 고유값과 고유벡터를 계산하지 않고도 식 (14) 를 이용하여 컴퓨터 타임을 근사적으로 구할 수 있다. 데이터 집합의 크기가 700과 2,000 인 패치 그래프에 대하여

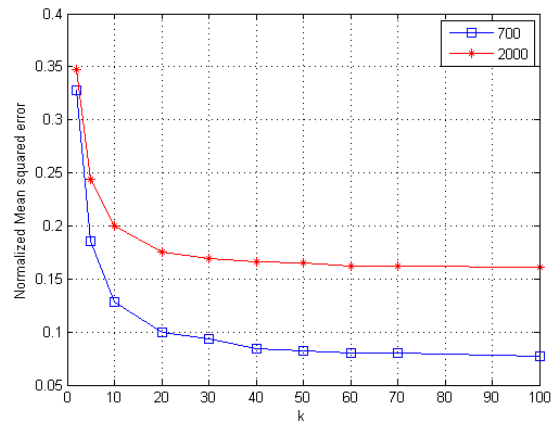


그림 3 사영행렬  $Q$  의 크기에 따른 ACTE의 NMSE  
파란 선: 700개 데이터, 붉은 선: 2,000개 데이터

Fig. 3. Normalized mean squared error as a function of the row size of the projection matrix  $Q$  of the ACTE, where the data sizes corresponding to the blue and red lines are 700 and 2,000, respectively.

사영 행렬  $Q$  를 각각 구한 다음,  $Q$  의 행 수에 따라 근사적으로 계산한 컴퓨터 타임의 정규 평균 제곱 오차(normalized mean squared error:  $NMSE$ ) 를 [그림 3.]에 제시하였다. 여기서  $NMSE$  는 다음과 같이 정의한다.

$$NMSE = \frac{\sum_{i \neq j}^n (\sqrt{c(x_i, x_j)} - \|Z(e_i - e_j)\|)^2}{\sum_{i \neq j}^n c(x_i, x_j)} \quad (16)$$

이 때, 보다 더 근사적인 결과를 얻으려면 식 (15)에서 알 수 있듯이 매개변수  $\epsilon$  에 작은 값을 할당하여야 하고 이에 따라  $k \times d$  랜덤 행렬  $Q$  가 결정된다. Nyström 방식의 근사화 방식은 표본화된 패치 수  $k$  에 대하여  $O(k^3)$  의 계산량이 요구되므로 허용 오차의 범위를 줄이기 위해서는 계산량이 급격히 증가하지만, 이 방식은 계산량이 거의 선형적으로 증가하므로 비교적 적은 계산량으로 보다 정확한 컴퓨터 타임 거리를 구할 수 있는 장점이 있다. 위의 두 방식은 컴퓨터 타임에 기반하고 있다는 점에서 상호 간에 공통점은 있으나, 이를 근사적으로 구하는 방식에 있어서는 구조적으로 다르므로 이들의 성능을 직접적으로 비교하기는 어려울 뿐만 아니라 이들의 응용분야에서도 많은 차이가 있다. 즉, Nyström 방식은 스펙트럼 분해 연산 과정이 필수적으로 요구되므로 데이터 집합의 크기가 제한적일 수밖에 없으나 ACTE는 그 크기에 영향을 받지 않는다. 한편, 전자는 주성분(principal components)을 쉽게 알 수 있으므로 데이터의 차원을 크게 줄일 수 있고 식 (6)에서와 같이 이차원 또는 삼차원 공간에 임베딩할 수 있어서 데이터의 기하구조를 시각화할 수 있는 장점이 있으나, 후자는 주성분을 알 수 없으므로 데이터의 차원을 줄이는데 어려움이 있을 뿐만 아니라 데이터의 시각화도 사실상 불가능한 한계를 보이는 특징이 있다.

## VI. 결 론

본 논문에서는 가장 널리 활용되고 있는 다양체 학습 방식인 컴퓨터 타임 임베딩을 구현하는데 요구되는 엄청난 계산량을 줄이기 위한 두가지 방안을 제시하고 실험을 통하여 이들의 성능을 확인하였다. 기본적으로 그래프의 가중치 행렬에서 랜덤으로 열을 표본화한 다음

이를 이용하여 정규 라플라시안 행렬을 구하면 더 이상 SPSPD하지 않고, 또한 가장 작은 고유값이 0이어야 한다는 조건을 만족시키지 못함으로 해서 근사화 오차가 크게 발생함을 피할 수 없다. 본 논문에서는 이러한 문제를 해결하기 위한 새로운 방식을 제안하였고 실험을 통하여 성능을 확인하였다. 하지만, 이 방식도 데이터 집합의 크기가 매우 크면, 라플라시안 행렬을 스펙트럼 분해하는데 계산량이 기하급수적으로 증가하므로 적용하기 어려운 단점이 있다. 이를 해결하기 위하여 표본화 과정과 라플라시안 행렬에 대한 고유값과 고유벡터의 계산과정 없이 입력 데이터 집합으로부터 컴퓨터 타임 거리를 직접 추정하는 방식을 이용하였다. 이 방식은 입력 데이터 집합의 크기에 영향을 받지 않으면서 컴퓨터 타임을 근사적으로 구할 수 있는 장점이 있다. 향후에는 데이터 집합의 크기가 증가함에 따라 근사화 오차가 비교적 크게 비례하여 나타나는데, 이의 원인을 분석하고 해결하는 연구를 수행할 계획이다.

## REFERENCES

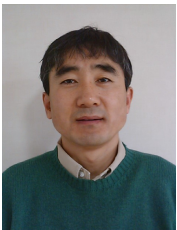
- [1] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences* 66, pp. 671-687, 2003.
- [2] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation* 15(6), pp.1373-1396, 2003.
- [3] A. Choromanska, T. Jebara, H. Kim, M. Mohan, and C. Monteleoni, "Fast Spectral clustering via the Nyström method," *ALT2013, LNAI 8139*, pp. 367-381, 2013.
- [4] F. Chung, *Spectral graph theory*, American Mathematical Society, 1997.
- [5] P. Drineas and M.W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *Journal of Machine Learning Research* 6, pp. 2153-2175, 2005.
- [6] H.I. Hahn, "Analysis of commute time embedding based on spectral graph," *Journal of Korea Multimedia Society, Vol. 17, No. 1*, pp. 34-42, 2013.
- [7] H.I. Hahn, "A Study on classification of waveforms using manifold embedding based on commute time," *Journal of the Institute of*



- Electronics and Information Engineers, Vol. 51, No. 2*, pp. 148-155, 2014.
- [8] H. Qiu and E.R. Hancock, "Clustering and embedding using commute times," *IEEE Trans. PAMI*, Vol. 29, No. 11, Nov., 2007.
- [9] S. T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science Vol.290*, 2000.
- [10] D. A. Spieman and N. Srivastava, "Graph sparsification by effective resistances," *In Proceedings of the 40<sup>th</sup> Annual ACM Symposium on Theory of Computing, STOC'04*, pp. 81-90, 2004.
- [11] D.A. Spieman and S. Teng, "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear system," arXiv:cs/0607105v5 [cs.NA], 2012.
- [12] J.B. Tenenbaum, V. deSilva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science, Vol. 290*, pp.2319-2323, 2000.
- [13] C.K.I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," *In Annual Advances in Neural Information Processing Systems 13: Proceeding of the 2000 Conference*, pp. 682-688, 2001.

---

— 저 자 소 개 —



한 희 일(평생회원)

1984년 서울대학교 제어계측  
공학과 학사 졸업.

1988년 서울대학교 제어계측  
공학과 석사 졸업.

1995년 University of Arizona  
전기및컴퓨터공학과  
박사 졸업.

2014년 현재 한국외국어대학교 정보통신공학과  
교수.

<주관심분야 : 신호처리, 컴퓨터비전, 패턴인식>