Efficient large scale commute time embedding

To cite this article: H I Hahn 2017 J. Phys.: Conf. Ser. 814 012010

View the article online for updates and enhancements.

Related content
- New method for computing eigenfunctions (Mathieu functions) for scattering by elliptical cylinders
  J J Stamnes and B Spjelkavik
- ON THE LAPLACIAN AND GAUSSIAN ORBIT METHODS
  S. Herrick
- Second order symmetries of the conformal laplacian and R-separation
  Jean-Philippe Michel, Fabian Radoux and Josef Silhan
Efficient large scale commute time embedding

H I Hahn

Department of Information and Communications Eng., Hankuk University of Foreign Studies, 89 Wangsan, Mohyun, Yongin, Kyonggi-Do, 449-791, Korea

hihahn@hufs.ac.kr

Abstract. Commute time embedding involves computing eigenfunctions of the graph Laplacian matrix. Spectral decomposition requires computational burden proportional to $O(n^3)$, which may not be suitable for large scale dataset. This paper proposes computationally efficient commute time embedding by applying Nyström method to the normalized graph Laplacian. The performance of the proposed algorithms is analysed by checking the embedding results on a patch graph.

1. Introduction

When dataset lies on a linear subspace, PCA(principal component analysis) is most useful and optimal for embedding as well as dimensionality reduction in terms of maintaining maximum variance of dataset. However, when dataset lies on a nonlinear space, PCA introduces severe error. The manifold learning algorithms replace PCA on a nonlinear space. Although there are lots of manifold learning algorithms, commute time embedding is known to be most suitable for dimensionality reduction.

Basically, manifold learning algorithms usually use kernel-based methods to represent the data of nonlinear structure, because it is more clear in the feature space rather than in the original input space. In kernel-based methods, kernel matrix is defined when input data are mapped to the feature space using a map $\Phi: \mathcal{X} \rightarrow \mathcal{F}$. Kernel matrix $G \in \mathbb{R}^{n \times n}$, whose element $G_{ij} = k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$, can be represented in terms of an inner product between feature vectors. For example, ISOMAP [1] computes the matrix whose elements are the geodesic distance between pairs of nodes. On the contrary, Laplacian eigenmap [2] and commute time embedding [3] employ graph Laplacian matrices as kernels. This scheme needs to compute the spectral decomposition of the kernel matrix, whose computational burden increases in the order of $O(n^3)$, where $n$ is the size of dataset. It is hardly applicable when $n$ is sufficiently large. Lots of researchers have tried to solve this problem by accelerating the computational time of spectral decomposition. Nyström method, proposed by Williams, et al. [4], is adopted in [5] to compute the kernel eigenfunctions approximately from the affinity matrix composed of randomly chosen samples. In spite of lots of researches on approximation of kernel matrices, there have been rarely known about approximation of the normalized graph Laplacian. The properties that Graph Laplacian matrix is symmetric and positive semidefinite (SPSD) should be preserved even after its approximation. However, the method proposed by Choromanska, et al. [6] does not satisfy it any more. They try to create a submatrix by random sampling the columns of graph Laplacian and normalizing it directly. However, it induces a severe error because the submatrix is no longer SPSD. This paper proposes a new method to reduce the approximation error significantly. Commute time embedding is performed on the patch graph, so that each patch is mapped into the corresponding vector.
2. Review of Commute Time Embedding

2.1. Construction of patch graph

Suppose that maximally overlapped patches of size $p$ samples are extracted around each time sample in the following way:

$$\hat{s}_n = \frac{\bar{s}_n}{\| \bar{s}_n \|}, \quad n = 1, 2, \ldots, N,$$

where $\bar{s}_n$ is the mean centered version of $s_n = (x[n], x[n+1], \ldots, x[n+p-1]) \in \mathbb{R}^p$. The patch $\hat{s}_n$ is obtained by normalizing with the magnitude. The weight along the edge connecting the nodes $v_i$ with $v_j$, which are associated with $\hat{s}_i$ and $\hat{s}_j$, respectively, is defined as follows:

$$w(i, j) = \begin{cases} \exp \left(-\frac{\| \hat{s}_i - \hat{s}_j \|^2}{2\sigma^2} \right) & v_i, v_j: \text{connected} \\ 0 & \text{otherwise.} \end{cases}$$

Given a set of patches $\hat{s}_1, \ldots, \hat{s}_N$ and $w(i, j)$, we can construct a graph, where two vertices $v_i$ and $v_j$ are connected with a weight $w(i, j)$.

2.2. Commute time embedding

Given an affinity matrix $W \in \mathbb{R}^{N \times N}$, whose entries are $w(u, v)$, the degree matrix $D$ is computed to be diagonal with entries $d_i = \sum_{v} w(u, v)$ and the graph Laplacian is defined as $L = D - W$. Let $L = U \Lambda U^T$ be the spectral decomposition of $L$, where $U$ comprises all eigenvectors and $\Lambda$ is diagonal with the corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. The Moore-Penrose inverse of $L$, $L' = U \Lambda' U^T$ is defined as

$$\lambda_i' = \begin{cases} 1/\lambda_i & \lambda_i \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

where $\Lambda' = \text{diag}(\lambda_1', \ldots, \lambda_n')$. Then, the commute time between two vertices $v_i$ and $v_j$ is given as

$$c(i, j) = \text{vol}(e_i, e_j)^T L' (e_i, e_j),$$

where $\text{vol} = \sum_v w(u, v)$ and $e_i = [0, \ldots, 1, 0, \ldots, 0]^T$ is defined as the $i^{th}$ column of the identity matrix $I$. Hence, $\sqrt{c(i, j)}$ can be considered a Mahalanobis distance with a weighting matrix $\text{vol} \cdot L'$, and $c(i, j)$ can be rephrased as

$$c(i, j) = \text{vol}(e_i, e_j)^T U \Lambda' U^T (e_i, e_j) = (z_i, z_j)^T (z_i, z_j),$$

where $z_i = \sqrt{\text{vol}} \left[ u_{i,1} / \sqrt{\lambda_1}, \ldots, u_{i,N} / \sqrt{\lambda_n} \right]$. Here $u_{i,j}$ is the $j^{th}$ element of $u_i$. $L$ can be normalized as $L_{sym} = D^{-1/2}LD^{-1/2}$. If we use $L_{sym}$ instead of $L$, $z_i$ can be rephrased as:

$$z_i = \sqrt{\text{vol}} \left[ \frac{v_{i,1}}{\sqrt{\lambda_1^2 d_i}}, \ldots, \frac{v_{i,N}}{\sqrt{\lambda_n^2 d_i}} \right],$$

where $\lambda_i'$ is the eigenvalue of $L_{sym}$ and $v_i$ is the corresponding eigenvector. For the dimensionality reduction, we can use only the first $q$ components corresponding to the lower eigenvectors. $\lambda_i'$'s are assumed to be sorted in the following way [3]:

$$0 \leq \lambda_1' \leq \lambda_2' \leq \cdots \leq \lambda_N' < 2.$$

3. Approximate commute time embedding using Nyström method

Let $C$ be the $N \times c$ matrix obtained by random sampling $c$ columns of an adjacency matrix $W$ uniformly without replacement and $A$ be the $c \times c$ matrix obtained by extracting the rows of the same indices as the sampled $c$ columns of $W$. The rows and columns of $W$ are rearranged as follows:
Since \( W \) is SPSD, \( A \) is also SPSD. Choromanska, et al. [6] approximate \( L_d = D^{-1/2}LD^{-1/2} \) by computing \( \hat{L}_d \leftarrow I - \sqrt{c/N}D^{-1/2} \cdot C \cdot A^{-1/2} \cdot \hat{D}^{-1/2} \). Here, \( \hat{I} \) is obtained by sampling the columns of the identity matrix \( I \) where \( \Omega \) represents the index set \( \{i_c \} \) sampled uniformly, which is denoted as \( \hat{I} \leftarrow I(\cdot, \Omega) \). And the degree matrices \( D \in \mathbb{R}^{N \times N}, \Delta \in \mathbb{R}^{c \times c} \) are defined as follows:

\[
D(i,i) = \sum_{j=1}^{c} C(i,j), \quad \Delta(i,i) = \sum_{j=1}^{N} C(j,i). \tag{8}
\]

They approximate the eigenvalues and eigenvectors of \( \hat{L}_d \) using the spectral decomposition of \( \hat{L}_d \leftarrow \hat{L}_d(\cdot, \cdot) \). The eigenvalues and eigenvectors of \( \hat{L}_d \) deviate from those of \( L_d \) and do not even satisfy the properties that the least eigenvalue be zero, since it is no longer SPSD.

In this paper, we propose a different method to approximate \( L_d \), which preserves the properties that \( L_d \) be SPSD even though it is approximated. At first, \( A \) is normalized using its degree matrix \( D_a \) to get \( \hat{A} \leftarrow D_a^{-1/2}AD_a^{-1/2} \), where \( D_a(i,i) = \sum_j A(i,j) \). However, \( B \) should be normalized differently, as given in the fourth step of the algorithm explained in figure 1, since \( B \) is neither a symmetric nor a square matrix. Finally, we approximate the eigenvalues and eigenvectors of \( L_d \) via the spectral decomposition of \( \hat{A} \), using the fact that the eigenvectors of \( W \) are the same as those of \( L_d \) and the relationship between the eigenvalues of \( W \) and \( L_d \) are \( \frac{1}{\lambda_{W}} = -\lambda_{L} \), where \( \Lambda_{W} \) and \( \Lambda \) are those of \( W \) and \( L_d \), respectively.

Figure 1 explains in detail the procedure to approximate \( L_d \) using Nyström method.

---

Proposed Nyström approximation of the normalized Laplacian \( L_d \)

**Input**: adjacency matrix \( W \in \mathbb{R}^{N \times N} \), the number of columns to be sampled \( c \)

**Output**: The approximate eigenvalues \( \hat{\Lambda} \) and eigenvectors \( \hat{U} \in \mathbb{R}^{N \times c} \) of \( L_d \)

1. \( C \leftarrow W(\cdot, \Omega) \)
2. \( A \leftarrow C(\Omega, \cdot), B \leftarrow C(\Omega^c, \cdot) \)
3. \( \hat{A} \leftarrow D_a^{-1/2}AD_a^{-1/2}, D_a(i,i) = \sum_j A(i,j) \)
4. \( \hat{B} \leftarrow D_b^{-1/2}BD_b^{-1/2}, \text{ where } D_b(i,i) = \sum_j A(i,j) \), \( \Lambda_b(i,i) = \frac{c}{N-c} \sum_j A(i,j) \)
5. **Spectral decomposition**: \( \hat{A} = U \Lambda \hat{U}^T \)
6. \( \hat{C} \leftarrow \begin{bmatrix} \hat{A} \\ \hat{B} \end{bmatrix}_{\Omega}, \text{ rearrange in the reverse order } \Omega. \)
7. \( \hat{\Lambda} = 1 - \frac{N}{c} \Lambda_b, \hat{U} = \frac{\sqrt{c/N} \hat{C} U \hat{\Lambda}^T}{\sqrt{c/N} \hat{C} U \hat{\Lambda}^T} \)

**Figure 1**. The proposed Nyström approximation of the normalized Laplacian.
4. Experiments
Patch sets associated with signals are constructed. Commute time embedding is performed on the patch set composed of \( N \) patches, each vertex is mapped into three dimensional vector, as shown in Equation (5) to get a feel for what the dataset looks like [7].

4.1. Investigating the characteristics of commute time embedding
In figure 2, we show commute time embedding of some sinusoid. The segment is composed of 700 samples, from which 676 patches are extracted so that they may be maximally overlapped. In this figure, patches of lower variance are encoded with blue colour, while patches of higher variance with red colour.

Figure 2. Commute time embedding of a sinusoidal signal from which the patch graph is constructed. Left: Two-dimensional Embedding, Right: Three-dimensional embedding.

As mentioned in the previous section, commute time embedding can be used as a tool for geometrization of dataset, where some periodic phenomena are transformed into the geometries composed of topological circles or higher dimensional holes in some space.

4.2. The proposed approximate commute time embedding using Nyström method
Figure 3 shows the commute time embedding results of the patch graph same as that of figure 2, using Nyström methods. As mentioned in the above section, severe distortion occurs on the embedding results of Choromanska’s method, compared with those of our proposed method. Irregular scales at the embeddings, which is caused by the errors in approximating the eigenvalues of \( L_{\text{sym}} \), seems to degrade the performance. Note that the phase can be reversed because the embedding map consists of eigenvectors, as given in equation (5). Table 1 shows the five smallest eigenvalues of the \( L_{\text{sym}} \) for the patch graph of sinusoid. According to equation (6), the smallest eigenvalue of \( L_{\text{sym}} \) for the connected graph should be zero and the largest one be less than 2. Choromanska’s method, as given in table 2, does not satisfy the conditions. However, table 3 shows that the approximate eigenvalues computed using our proposed method approach true values as the number of samples increases as well as satisfy the condition given in equation (6).

**Table 1.** The five smallest eigenvalues of \( L_{\text{sym}} \).

<table>
<thead>
<tr>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_3 )</th>
<th>( \lambda_4 )</th>
<th>( \lambda_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>0</td>
<td>0.01</td>
<td>0.012</td>
<td>0.040</td>
</tr>
</tbody>
</table>

**Table 2.** The five smallest eigenvalues of \( L_{\text{sym}} \), obtained using the Choromanska’s method.

<table>
<thead>
<tr>
<th>#samples</th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_3 )</th>
<th>( \lambda_4 )</th>
<th>( \lambda_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>2.304</td>
<td>2.331</td>
<td>2.340</td>
<td>2.359</td>
<td>2.425</td>
</tr>
<tr>
<td>400</td>
<td>0.637</td>
<td>0.650</td>
<td>0.711</td>
<td>0.729</td>
<td>0.745</td>
</tr>
<tr>
<td>600</td>
<td>0.119</td>
<td>0.131</td>
<td>0.136</td>
<td>0.166</td>
<td>0.175</td>
</tr>
</tbody>
</table>
Figure 3. Approximate commute time embedding using Nyström method. The upper row: Choromanska’s method, the lower row: our proposed method, Sampling rates: The first column (29.6%), the second column (59.2%), the third column (88.7%).

Table 3. The five smallest eigenvalues of $L_{nm}$, obtained using our proposed method.

<table>
<thead>
<tr>
<th>#samples</th>
<th>$\lambda_1'$</th>
<th>$\lambda_2'$</th>
<th>$\lambda_3'$</th>
<th>$\lambda_4'$</th>
<th>$\lambda_5'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.0</td>
<td>0.006</td>
<td>0.009</td>
<td>0.027</td>
<td>0.036</td>
</tr>
<tr>
<td>400</td>
<td>0.0</td>
<td>0.009</td>
<td>0.011</td>
<td>0.038</td>
<td>0.041</td>
</tr>
<tr>
<td>600</td>
<td>0.0</td>
<td>0.010</td>
<td>0.011</td>
<td>0.040</td>
<td>0.046</td>
</tr>
</tbody>
</table>

5. Conclusions

This paper has explored approximation of commute time embedding to reduce the computational burden for large dataset. Our proposed method is based on Nyström sampling method to compute approximate commute time. The strength of our method is that it preserves the properties that the normalized graph Laplacian matrix is SPSD even though it is approximated via sampling process. Thus, our method can be applied efficiently to dimensionality reduction, which is very effective for visualization, as well as spectral clustering or pattern classification of large dataset.

As a future research, we would like to explore its application to pattern classification or manifold visualization in a geometric way.

6. References