Hashing with Generalized Nyström Approximation

Jeong-Min Yun[1], Saehoon Kim[1], Seungjin Choi[1,2,3]
[1] *Department of Computer Science and Engineering,*
[2] *Division of IT Convergence Engineering,*
[3] *Department of Creative IT Excellence Engineering,*
Pohang University of Science and Technology, Pohang 790-784, Korea
Email: {azida,kshkawa,seungjin}@postech.ac.kr

***Abstract*—Hashing, which involves learning binary codes to embed high-dimensional data into a similarity-preserving low-dimensional Hamming space, is often formulated as linear dimensionality reduction followed by binary quantization. Linear dimensionality reduction, based on maximum variance formulation, requires leading eigenvectors of data covariance or graph Laplacian matrix. Computing leading singular vectors or eigenvectors in the case of high-dimension and large sample size, is a main bottleneck in most of data-driven hashing methods. In this paper we address the use of generalized Nyström method where a subset of rows and columns are used to approximately compute leading singular vectors of the data matrix, in order to improve the scalability of hashing methods in the case of high-dimensional data with large sample size. Especially we validate the useful behavior of *generalized Nyström approximation with uniform sampling*, in the case of a recently-developed hashing method based on principal component analysis (PCA) followed by an iterative quantization, referred to as PCA+ITQ, developed by Gong and Lazebnik. We compare the performance of generalized Nyström approximation with uniform and non-uniform sampling, to the full singular value decomposition (SVD) method, confirming that the uniform sampling improves the computational and space complexities dramatically, while the performance is not much sacrificed. In addition we present *low-rank approximation error bounds* for generalized Nyström approximation with uniform sampling, which is not a trivial extension of available results on the non-uniform sampling case.**

***Keywords*-CUR decomposition; hashing; generalized Nyström approximation; pseudoskeleton approximation; uniform sampling;**

I. Introduction

Hashing refers to methods for embedding high-dimensional data into a low-dimensional Hamming space such that similar data points are indexed by binary codes with small Hamming distances [1], [2]. A variety of hashing methods have been developed, since hashing has been shown to be better suited to approximate similarity search, especially for high-dimensional data, compared to tree structure-exploited methods [3]. An earlier work on notable data-independent method is *locality sensitive hashing* (LSH) [1] where random projections followed by rounding are used to generate binary codes such that two objects in database within a smaller distance are shown to have a higher probability of collision. However, for successful performance in real-world applications, LSH usually requires longer binary codes or multiple hash tables [1].

Data-dependent hashing, which is also known as 'learning to hash' seeks a similarity-preserving embedding into a Hamming space, where compact binary codes for efficient indexing are learned from a set of training examples. Spectral hashing (SH) [2] is a representative unsupervised hashing method, where a subset of eigenvectors of graph Laplacian matrix is rounded to determine binary codes. Semantic hashing [4] uses multi layers of restricted Boltzman machines to learn a non-linear mapping between input data and binary code bits in supervised manner. Semi-supervised learning, which uses both plenty of unlabeled examples and small number of labeled examples, has also been applied to hashing [5], [6], where hash function is mainly learned from labeled data while unlabeled data are used for regularization.

Many data-dependent hashing methods employ linear dimensionality reduction as an initial step, in order to project high-dimensional data into a low-dimensional subspace, which often requires the computation of leading eigenvectors of data covariance matrix [7] or graph Laplacian matrix [2]. Low-dimensional projection is followed by binary quantization to yield compact binary codes. Subspace computed by principal component analysis (PCA) was shown to be a promising solution to low-dimensional projection of hashing in information-theoretical perspective [7].

One of state of the arts along this direction, which is also of our interest in this paper, is PCA+ITQ [8], where one project data points into principal subspace and determine binary codes by iteratively minimizing the quantization error of mapping the projected data to the vertices of a zero-centered binary hypercube. The scalability of PCA+ITQ is limited by the computational and space complexities required to compute leading singular vectors of the data matrix. For instance, in the case of a dataset which contains hundreds of thousands of samples with each sample represented by hundreds of attributes, all data points are fit in 24 Gbytes memory, so full SVD is affordable. However, for larger-scale dataset, such as Holidays + Flickr1M dataset [9] which contains a million of samples with each sample corresponding to ten thousand-dimensional vector, about 90 Gbytes is required to store all data points. A usual standard machine with 32 Gbytes memory cannot afford to use the full SVD. PCA+ITQ becomes quickly impractical as the size of dataset grows, although hashing has been developed to handle large-scale data. To overcome this limitation, we employ a sampling-based method to approximately compute leading singular vectors of a large-scale data matrix using a subset of columns and rows.

Sampling-based methods are attractive and powerful techniques for approximately computing SVD or spectral decomposition, since they operate on a small block of the original matrix. Column-sampling method [10] approximates the SVD of a rectangular matrix by using the SVD of a small block of sampled columns. Nyström method [11] provides a low-rank approximation of a symmetric positive semi-definite matrix, making use of the SVD of the intersection of sampled columns with the corresponding rows of the original matrix. Recent studies in [12] reveal the similarities and relative advantages between these two methods. We refer to the extension of Nyström method to a rectangular matrix as *generalized Nyström method*, which is also known as *pseudoskeleton approximation* [13] or *CUR decomposition* [14], [15], [16]. It uses only a subset of columns and rows to approximate a rectangular matrix.

The sampling strategy is an important component of sampling-based methods. Nyström method with uniform sampling without replacement has shown to be quite efficient both in time and space in practice and its performance bound on approximation error was derived [17]. Nyström method with non-uniform sampling, where columns are sampled from a fixed distribution was also studied in [18]. The performance of CUR decomposition with non-uniform sampling was analyzed in [14], [15], [16]. In this paper we present bounds on approximation error, for generalized Nyström method, when used with uniform sampling without replacement. This analysis is motivated by seminar work on CUR decomposition [14], [15], [16], but our results are not trivial extension of existing work.

The main contribution of this paper is two fold: (1) the use of generalized Nyström method in PCA+ITQ to improve the scalability in both time and space; (2) approximation error bound for generalized Nyström method with uniform sampling. Experiments on several widely used benchmark datasets demonstrate that the performance of generalized Nyström approximation with uniform sampling is not much sacrificed in PCA+ITQ, while improving the computational and space complexities dramatically over the full SVD as well as non-uniform sampling method.

II. Generalized Nyström Approximation

A. Notation

Given $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and assuming w.l.o.g. that $m \le n$, the SVD of $\boldsymbol{X}$ is denoted by $\boldsymbol{X} = \boldsymbol{U}_X \boldsymbol{\Sigma}_X \boldsymbol{V}_X^\top$, where $\boldsymbol{X}_k = \boldsymbol{U}_{X,k}\boldsymbol{\Sigma}_{X,k}\boldsymbol{V}_{X,k}^\top$ corresponds to rank-k approximation using SVD. It is known that $\boldsymbol{X}_k = \arg\min_{\text{rank}(\widehat{X})\le k} \|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F$. We define a complementary matrix $\boldsymbol{X}_k^\perp = \boldsymbol{X} - \boldsymbol{X}_k = \boldsymbol{U}_{X,k}^\perp \boldsymbol{\Sigma}_{X,k}^\perp \boldsymbol{V}_{X,k}^{\perp\top}$, and the Moore-Penrose pseudoinverse $\boldsymbol{X}^+ = \boldsymbol{V}_X \boldsymbol{\Sigma}_X^+ \boldsymbol{U}_X^\top$. We denote $\boldsymbol{x}_i$ as the i-th column vector of $\boldsymbol{X}$, and $\boldsymbol{x}^j$ as the j-th row vector of $\boldsymbol{X}$.

We also use the following sampling matrices [15]: Let c and r be the number of sampled columns and rows respectively, then a column sampling matrix is defined by $\mathcal{S}_C = \sqrt{n/c}\boldsymbol{S}_C \in \mathbb{R}^{n\times c}$, where $[\boldsymbol{S}_C]_{ij} = 1$ if $\boldsymbol{x}_i$ is selected in the j-th trial, 0 otherwise. Similarly, $\mathcal{S}_R = \sqrt{m/r}\boldsymbol{S}_R \in \mathbb{R}^{r\times m}$ can be defined. Using these, we define three different scaled sub-matrices of $\boldsymbol{X}$: $\boldsymbol{C} = \boldsymbol{X}\mathcal{S}_C \in \mathbb{R}^{m\times c}$, $\boldsymbol{R} = \mathcal{S}_R\boldsymbol{X} \in \mathbb{R}^{r\times n}$, and $\boldsymbol{W} = \mathcal{S}_R\boldsymbol{X}\mathcal{S}_C \in \mathbb{R}^{r\times c}$.

B. Generalized Nyström Method

Given a symmetric positive semi-definite (SPSD) matrix $\boldsymbol{G} \in \mathbb{R}^{n\times n}$, the standard Nyström method [11] is popular as an approximate algorithm of eigenvectors and eigenvalues of $\boldsymbol{G}$. It firstly builds two submatrices of $\boldsymbol{G}$: $\boldsymbol{C}_G = \boldsymbol{G}\mathcal{S}_C$ and $\boldsymbol{W}_G = \mathcal{S}_C^\top \boldsymbol{G}\mathcal{S}_C$, then constructs a k-rank approximation of $\boldsymbol{G}$ as follows: $\boldsymbol{G} \approx \widetilde{\boldsymbol{G}}_k = \boldsymbol{C}_G \boldsymbol{W}_{G,k}^+ \boldsymbol{C}_G^\top$. Approximate eigenvectors and eigenvalues are $\widetilde{\boldsymbol{U}}_G = \boldsymbol{C}_G \boldsymbol{U}_{W_G,k}\boldsymbol{\Sigma}_{W_G,k}^+$ and $\widetilde{\boldsymbol{\Sigma}}_G = \boldsymbol{\Sigma}_{W_G,k}$ respectively.

For an arbitrary matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, we consider a direct generalization of the standard Nyström method [13] like this: With $\boldsymbol{C} = \boldsymbol{X}\mathcal{S}_C$, $\boldsymbol{R} = \mathcal{S}_R\boldsymbol{X}$, and $\boldsymbol{W} = \mathcal{S}_R\boldsymbol{X}\mathcal{S}_C$,

$$\boldsymbol{X} \approx \widetilde{\boldsymbol{X}}_k = \boldsymbol{C}\boldsymbol{W}_k^+\boldsymbol{R} = \boldsymbol{C}\boldsymbol{V}_{W,k}\boldsymbol{\Sigma}_{W.k}^+\boldsymbol{U}_{W,k}^\top\boldsymbol{R}, \tag{1}$$

where approximate left singular vectors, right singular vectors, and singular values are $\widetilde{\boldsymbol{U}}_X = \boldsymbol{C}\boldsymbol{V}_{W,k}\boldsymbol{\Sigma}_{W,k}^+$, $\widetilde{\boldsymbol{V}}_X = \boldsymbol{R}^\top\boldsymbol{U}_{W,k}\boldsymbol{\Sigma}_{W,k}^+$, and $\widetilde{\boldsymbol{\Sigma}}_X = \boldsymbol{\Sigma}_{W,k}$, respectively. From this, you can see that $\widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{\Sigma}}_X\widetilde{\boldsymbol{V}}_X^\top = \boldsymbol{C}\boldsymbol{W}_k^+\boldsymbol{R} \approx \boldsymbol{X}$.

Without considering the sampling time of columns and rows of $\boldsymbol{X}$, the computation times of $\widetilde{\boldsymbol{U}}_X$, $\widetilde{\boldsymbol{V}}_X$, and $\widetilde{\boldsymbol{\Sigma}}_X$ are $O(mck + \min\{c^2r, cr^2\})$, $O(nrk + \min\{c^2r, cr^2\})$, and $O(\min\{c^2r, cr^2\})$ respectively. The reconstruction of $\boldsymbol{X}$ requires $O(mnk + \min\{c^2r, cr^2\})$ time. For the sampling time, if we use the uniform sampling without replacement strategy, the algorithm takes additional $O(m+n)$ time, whereas it needs $O(mn)$ additional time with the non-uniform sampling strategy.

C. Previous Error Bound Analysis of Generalized Nyström

Generalized Nyström method is designed to approximate some components of $\boldsymbol{X}_k = \boldsymbol{U}_{X,k}\boldsymbol{\Sigma}_{X,k}\boldsymbol{V}_{X,k}^\top$, and its quality is usually measured by the low-rank approximation error: $\|\boldsymbol{X} - \widetilde{\boldsymbol{X}}_k\|_F$ after forming a low-rank approximation $\widetilde{\boldsymbol{X}}_k$ using the approximated components. Based on how many components of $\boldsymbol{X}_k$ are approximated, there are two ways to form $\widetilde{\boldsymbol{X}}_k$: If only $\boldsymbol{U}_{X,k}$ is approximated as $\widetilde{\boldsymbol{U}}_X$, we form $\widetilde{\boldsymbol{X}}_k$ as $\widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X^\top\boldsymbol{X}$, which is an approximate projection of

$\boldsymbol{X}$ onto the subspace spanned by $\widetilde{\boldsymbol{U}}_X$[1]. We call this 'matrix projection' [12]. If all three components are available, $\widetilde{\boldsymbol{X}}_k$ can be formed by $\widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{\Sigma}}_X\widetilde{\boldsymbol{V}}_X^\top$, and we refer this to 'matrix reconstruction'. As you will see in Section III-B, the algorithm only uses $\widetilde{\boldsymbol{U}}_X$, so our theoretical justification about its quality is based on 'matrix projection'.

For matrix projection, there are two theoretical studies about the low-rank approximation error bound with the Frobenius norm [10], [14]. Both methods basically sample rows and columns based on the non-uniform distribution, and show that the resulted $\widetilde{\boldsymbol{X}}_k$ satisfies

$$\|\boldsymbol{X}-\widetilde{\boldsymbol{X}}_k\|_F \leq \|\boldsymbol{X}-\boldsymbol{X}_k\|_F + \epsilon\|\boldsymbol{X}\|_F \tag{2}$$

with high probability. The main difference of those is the required number of rows and columns to achieve the above bound. [10] requires $\Omega(k^2/\epsilon^4)$, and [14] requires $\Omega(\max\{k^4/c^3, k^2/\epsilon^4\})$.

III. Hashing with Generalized Nyström

In this section, we explain why principal subspace is useful for data-dependent hashing, and suggest the generalized Nyström approximation scheme, especially with uniform sampling to handle large-scale high-dimensional data.

A. PCA-based Hashing

Given a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_n] \in \mathbb{R}^{m\times n}$, hashing aims to find a hash function $\boldsymbol{h} : \mathbb{R}^m \rightarrow \{+1,-1\}^k$ with $k << m$, such that for all $i, j \in \{1,\dots,n\}$, Hamming distance between $\boldsymbol{h}(\boldsymbol{x}_i)$ and $\boldsymbol{h}(\boldsymbol{x}_j)$ is small if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are semantically similar. We use $\boldsymbol{h}_i(\cdot)$ to represent an i-th entry of the vector $\boldsymbol{h}(\cdot)$, and define $\boldsymbol{Y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_n] \in \{+1,-1\}^{k\times n}$, where $\boldsymbol{y}_i = \boldsymbol{h}(\boldsymbol{x}_i)$

To produce compact binary codes, data-dependent hashing mostly requires two conditions in which both are designed to maximize the information from the hash bits [2]:

- Balanced condition, $\sum_{i=1}^n \boldsymbol{h}_j(\boldsymbol{x}_i) = 0$, is a solution of an entropy maximization problem of the j-th hash bit $(\max \boldsymbol{H}[\boldsymbol{h}_j(\boldsymbol{x})])$ [7].
- Uncorrelated condition, $n^{-1}\boldsymbol{Y}\boldsymbol{Y}^\top = \boldsymbol{I}$, enforces that the hash bits are uncorrelated each other, so that there is no redundancy between the hash bits.

However, finding a hash function which satisfies the above conditions is an NP-hard problem [2], so some kind of relaxation of them is needed for practical use.

With a restriction of the form of the hash function; for all $j \in \{1,\dots,k\}$, $\boldsymbol{h}_j(\boldsymbol{x}) = \text{sgn}(\boldsymbol{u}_j^\top\boldsymbol{x})$ where $\boldsymbol{u}_j \in \mathbb{R}^m$ and $\text{sgn}(z) = \frac{z}{|z|}$, [7] shows that the balanced condition is lower-bounded by the scaled variance of the projected data; $\max H[\boldsymbol{h}_j(\boldsymbol{x})] \geq c \cdot \text{var}[\boldsymbol{u}_j^\top\boldsymbol{x}]$. And by relaxing the uncorrelated condition to the orthogonal constraint as $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$, where $\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_k]$, they suggest the following optimization problem for the hash function learning:

$$\underset{\boldsymbol{U}}{\arg\max} \sum_{i=1}^n\sum_{j=1}^k \text{var}[\boldsymbol{u}_j^\top\boldsymbol{x}_i], \text{where } \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}. \tag{3}$$

This problem is equivalent to PCA, and the binary codes are obtained by applying $\text{sgn}(\cdot)$ to each entry of the projected data $\boldsymbol{U}^\top\boldsymbol{X}$. However, applying $\text{sgn}(\cdot)$ directly to this PCA-projection is problematic in practice. In the relaxed optimization problem, just a summation of variance of hash bits is maximized, but the actually good hash functions should have enough information (variance) for each hash bit. For the PCA of real-world datasets, only top few eigenvectors have a large variance, so that most hash functions are obtained from the directions that have a small variance; as a result, the performance is degraded as the code length increases [5], [8]. To avoid this situation, [5] adds a penalty term to the objective function for the orthogonality instead of the constraint. [8] proposes an iterative quantization (ITQ) method, which iteratively minimizes the quantization error of $\text{sgn}(\cdot)$ by rotating principal components.

Algorithm 1 PCA + ITQ with generalized Nyström approximation (uniform sampling)

```
Input: training data is $X = [x_1 \cdots x_n] \in \mathbb{R}^{m\times n}$, test data
    is $x \in \mathbb{R}^m$, and binary code length is $k$. $c$ and $r$ are the
    number of sampled columns and rows respectively.
Output: binary code $y$ associated with $x$
Sample $c$ columns of $X$ uniformly at random as
    $x_{t_1},\dots,x_{t_c}$, and form $C = \sqrt{\frac{n}{c}}[x_{t_1} \cdots x_{t_c}]$.
Sample $r$ rows of $C$ uniformly at random as
    $c^{t_1},\dots,c^{t_r}$, then $W = \sqrt{\frac{m}{r}}[(c^{t_1})^\top \cdots (c^{t_r})^\top]^\top$.
Apply SVD to $W$ and obtain $W_k = U_{W,k}\Sigma_{W,k}V_{W,k}^\top$.
Approximate left singular vectors $\widetilde{U}_X = CV_{W,k}\Sigma_{W,k}^+$.
$Q_0$ is a random rotation matrix.
for $i = 1,\dots,50$ do
    $Y_i = \text{sgn}(Q_{i-1}^\top\widetilde{U}_X^\top X)$.
    $Q_i = \widehat{M}M^\top$, where $Y_iX^\top\widetilde{U}_X$ is decomposed by
        SVD as $M\Omega\widehat{M}^\top$.
end for
Return $k$-bit binary code: $y = \text{sgn}(Q_{50}^\top\widetilde{U}_X^\top x)$.
```

B. PCA + ITQ with Generalized Nyström Approximation

Although PCA + ITQ provides a promising solution in many cases, PCA is very time-consuming for a large-scale high-dimensional dataset. So we suggest the generalized Nyström method with uniform sampling to approximate PCA, whose time complexity is only linear at m. Because the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^\top$ is equivalent to the left singular vectors of $\boldsymbol{X}$, we use the approximate left singular vector of $\boldsymbol{X}$ from the generalized Nyström method as an eigenvector

[1] We call this an 'approximate' projection since $\widetilde{\boldsymbol{U}}_X^2 = \widetilde{\boldsymbol{U}}_X$, the definition of the projection, may not be satisfied depending on the approximation algorithms.

approximation of $\boldsymbol{X}\boldsymbol{X}^{\top}$. The overall procedure for the uniform sampling case is shown in Algorithm 1.

IV. Experiments

In this experiment, we compare our generalized Nyström with uniform sampling + ITQ (Uniform + ITQ) to full PCA (full PCA + ITQ), generalized Nyström with non-uniform sampling (Non-Uniform + ITQ), and random projection + ITQ (Random + ITQ). For full PCA, we compute the data covariance and extract the top-k eigenvectors. For generalized Nyström with non-uniform sampling, we implement the ConstantTimeSVD algorithm in [14]. For random projection, we use the standard normal random matrix.

We use three widely used benchmark datasets: CIFAR-10 [19], NUS-WIDE [20], and Holidays + Flickr1M [9]. Although CIFAR-10 (60K × 1584) and NUS-WIDE (270K × 634) are small-scale so that full PCA could be performed very fast, we include them to check that Uniform + ITQ even works in the small scale. Holidays + Flickr1M consists of 1M images crawled from Flickr[2] (Flickr1M) and 500 queries with a few number of ground-truth images for each (Holidays). We extract 12,800-dimensional vectors for each image using VLAD descriptor [9]. All experiments are conducted five times using Intel i7 with 32 Gbytes memory. Matrix operations are implemented by Matlab, and disk I/O operation (data loading) and the selection algorithm for non-uniform sampling are implemented by C++. For evaluation, we compute mean average precision (mAP) as in [8]. For Holidays + Flickr1M, we compute mAP in the following way since they provide only the small number of ground-truth matches of each queries (usually 2 or 3). For each query, we retrieve 100 candidates (or slightly more if there is a tie) using hash bits, and compute an average precision from the re-sorted candidates; re-sorting is based on the projected data matrix from each method, since sorting with the original features needs additional disk I/O operation.

Fig. 1 represents mAP comparison over hash bits of various hashing methods with fixed sampling ratios, 5% for samples and 30% for features. We observe that approximate principle components, from both Uniform + ITQ and Non-Uniform + ITQ, are enough to learn an effective hash function, using only a small subset of samples and features. We also observe that the hashing performance of Uniform + ITQ is definitely superior to Random + ITQ since random projection does not approximate principle components. Fig. 2 compares mAP and the computation time over various sampling ratios on Holidays + Flickr1M dataset. As you can see in (a) and (b), mAP of Uniform + ITQ and Non-Uniform + ITQ become comparable to full PCA + ITQ with the sampling ratios larger than 5% for samples and 30% for features. In (c) and (d), we see that the computation time of Uniform + ITQ is at least twice faster than full PCA and Non-Uniform + ITQ, and as fast as random projection. Specifically with 256 hash bits, full PCA + ITQ takes about 4,000 sec for PCA, 1,500 sec for data projection, and 400 sec for ITQ. Note that data projection and ITQ are also needed for other methods. In case of 5% samples, Uniform + ITQ takes about 250 sec for PCA approximation, while Non-Uniform + ITQ needs additional time for the sampling probability computation (about 1,100 sec) and the selection algorithm (about 520 sec).

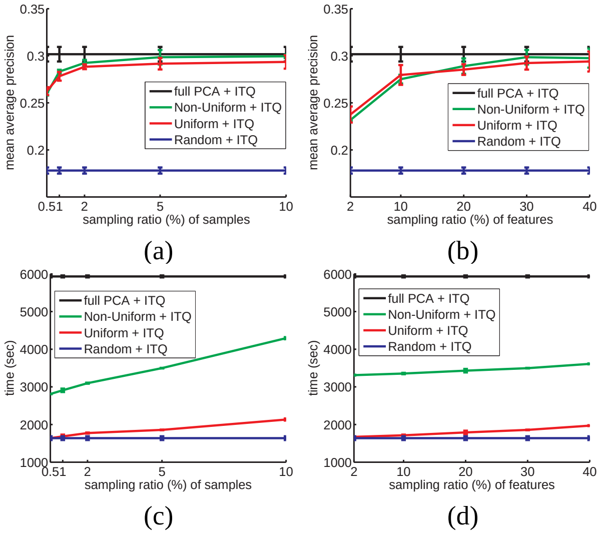

Figure 2. mAP (the first row) and the computation time (the second row) over various ratios (%) of samples and features on Holidays + Flickr1M. Each hashing methods use 256 hash bits. The sampling ratio of features is 30% in the left column, and that of samples is 5% in the right column.

V. Approximation Error Bound for Generalized Nyström with Uniform Sampling

Here, we provide the approximation error bound for matrix projection of $\boldsymbol{X}$ when the generalized Nyström method with uniform sampling without replacement is applied.

A. Concentration Bound for Matrix Multiplication

To derive our bound, we use the concentration bound for approximate matrix multiplication [17]:

Theorem 1. *(Theorem 2 of [17]). Given* $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ *and* $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$*, define* $\boldsymbol{C} = \boldsymbol{X}\mathcal{S}_C$*,* $\boldsymbol{R} = \mathcal{S}_C^{\top}\boldsymbol{Y}$*. Let* $\delta \in (0,1)$*,* $\alpha(u,v) = \frac{uv}{u+v-1/2}\frac{1}{1-1/(2\max\{u,v\})}$*, and* $\eta = \sqrt{\frac{2\log(2/\delta)\alpha(c,n-c)}{c}}$*. Then, the following holds*

$$\|\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{C}\boldsymbol{R}\|_F \leq \frac{(1+\eta)n}{\sqrt{c}} \max_i \|\boldsymbol{x}_i\| \max_j \|\boldsymbol{y}^j\|, \qquad (4)$$

with probability at least $1-\delta$*. If* $\boldsymbol{Y} = \boldsymbol{X}^{\top}$*, the above inequality satisfies with* $\eta = \sqrt{\frac{\log(2/\delta)\alpha(c,n-c)}{c}}$*.*

We also use the spectral norm version of Theorem 1:

[2] http://www.flickr.com/

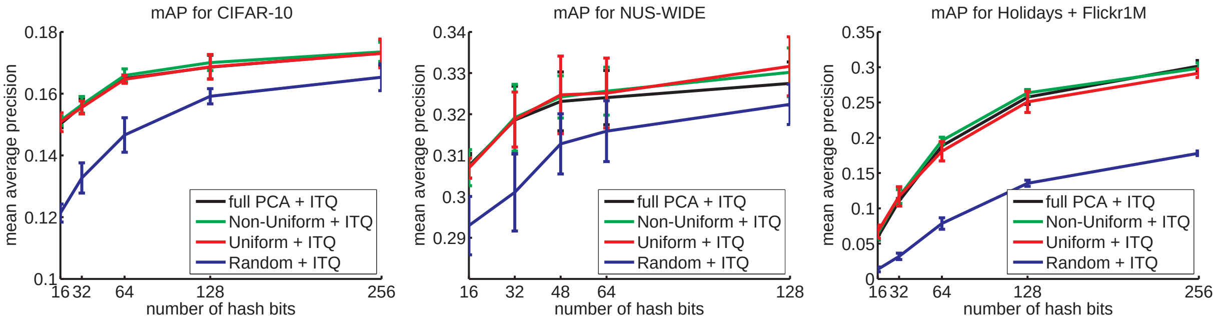

Figure 1. mAP comparison of various hashing methods. For Non-Uniform + ITQ and Uniform + ITQ, we fix the sampling ratios of samples and features to 5% and 30% respectively. For all figures, standard deviation (n = 5) are represented as error bars.

Corollary 1. *Let $\delta \in (0,1)$, $\eta = \sqrt{\frac{2\log(2/\delta)\alpha(c,n-c)}{c}}$. Then, with probability at least $1-\delta$,*

$$\|\boldsymbol{XY} - \boldsymbol{CR}\|_F \le \frac{(1+\eta)n}{\sqrt{c}} \|\boldsymbol{X}\|_2 \|\boldsymbol{Y}\|_2, \tag{5}$$

B. Modification of Generalized Nyström Approximation

We add a simple truncation step to the original method, ideas from the ConstantTimeSVD algorithm in [14], to bound $\boldsymbol{W}_k^+$. For $\gamma > 0$, We first compute SVD of $\boldsymbol{W}$ as $\boldsymbol{W}_k = \boldsymbol{U}_{W,k}\boldsymbol{\Sigma}_{W,k}\boldsymbol{V}_{W,k}$, and let $k' = \min\{k, |\{i : \sigma_i^2(\boldsymbol{W}_k) \ge \gamma\|\boldsymbol{C}\|_1\|\boldsymbol{C}\|_\infty\}|\}$. Then, a modified generalized Nyström approximation of $\boldsymbol{A}$ is

$$\boldsymbol{X} \approx \widetilde{\boldsymbol{X}}_k = \boldsymbol{C}\boldsymbol{W}_{k'}^+\boldsymbol{R} = \boldsymbol{C}\boldsymbol{V}_{W,k'}\boldsymbol{\Sigma}_{W,k'}^+\boldsymbol{U}_{W,k'}^\top\boldsymbol{R}. \tag{6}$$

C. Main Theorem

The following theorem is our main result.

Theorem 2. *Given a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. Let $\delta \in (0,1)$, $\eta_1 = \sqrt{\frac{\log(4/\delta)\alpha(c,n-c)}{c}}$, and $\eta_2 = \sqrt{\frac{2\log(4/\delta)\alpha(r,m-r)}{r}}$. For $\epsilon_1, \epsilon_2 > 0$, if $c \ge 4k/\epsilon_1^2$ and $r \ge 1/(\gamma^2\epsilon_2^2)$, then the modified Generalized Nyström method with uniform sampling without replacement produces $\widetilde{\boldsymbol{U}}_X$ which satisfy*

$$\|\boldsymbol{X} - \widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X\boldsymbol{X}\|_F \le \|\boldsymbol{X} - \boldsymbol{X}_k\|_F + \sqrt{n\epsilon_1(1+\eta_1)}\max_i\|\boldsymbol{x}_i\| + \{m\epsilon_2(1+\eta_2) + \mathcal{E}\}\|\boldsymbol{X}\|_2$$

with prob. at least $1-\delta$, where $\mathcal{E} = \frac{\sqrt{2}\|\boldsymbol{C}_k^\perp\|_F}{\sqrt{\gamma}\|\boldsymbol{C}\|_2} + \sqrt{k-k'}$.

Direct proof is not easy, so we divide $\|\boldsymbol{X} - \widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X\boldsymbol{X}\|_F$ into two subproblems using the triangle inequality as $\|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X}\|_F + \|\boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X} - \widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X^\top\boldsymbol{X}\|_F$, and derive an upper bound for each as follows.

1) Upper Bound for $\|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X}\|_F$: If we use Theorem 1 (concentration bound) and Theorem 2 of [14], we can directly derive the following low-rank approximation error bound from the fact that $\boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X} = \boldsymbol{U}_{C,k}\boldsymbol{U}_{C,k}^\top\boldsymbol{X}$:

Theorem 3. *Given a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. Let $\delta \in (0,1)$ and $\eta = \sqrt{\frac{\log(2/\delta)\alpha(c,n-c)}{c}}$. For $\epsilon > 0$, if $c \ge 4k/\epsilon^2$, then with probability at least $1-\delta$,*

$$\|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X}\|_F \le \|\boldsymbol{X} - \boldsymbol{X}_k\|_F + \sqrt{n\epsilon(1+\eta)}\max_i\|\boldsymbol{x}_i\|.$$

2) Upper Bound for $\|\boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X} - \widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X^\top\boldsymbol{X}\|_F$:

Lemma 1. $\|\boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X} - \widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X^\top\boldsymbol{X}\|_F$

$$\le \left[\frac{1}{\gamma\|\boldsymbol{C}\|_2^2}\|\boldsymbol{C}^\top\boldsymbol{C} - \boldsymbol{W}^\top\boldsymbol{W}\|_F + \mathcal{E}\right]\|\boldsymbol{X}\|_2, \tag{7}$$

where $\mathcal{E} = \mathcal{E}(\boldsymbol{C},\gamma,k,k') = \frac{\sqrt{2}\|\boldsymbol{C}_k^\perp\|_F}{\sqrt{\gamma}\|\boldsymbol{C}\|_2} + \sqrt{k-k'}$.

Proof: Let $\boldsymbol{Y} = \boldsymbol{\Sigma}_C\boldsymbol{V}_C^\top\boldsymbol{V}_{W,k'}\boldsymbol{\Sigma}_{W,k'}^+$ and $\boldsymbol{Y}_k = \boldsymbol{\Sigma}_{C,k}\boldsymbol{V}_{C,k}^\top\boldsymbol{V}_{W,k'}\boldsymbol{\Sigma}_{W,k'}^+$. Then, $\|\boldsymbol{C}\boldsymbol{C}_k^+\boldsymbol{X} - \widetilde{\boldsymbol{U}}_X\widetilde{\boldsymbol{U}}_X^\top\boldsymbol{X}\|_F$

$$\begin{aligned} &= \|\boldsymbol{U}_{C,k}\boldsymbol{U}_{C,k}^\top\boldsymbol{X} - \boldsymbol{U}_C\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{U}_C^\top\boldsymbol{X}\|_F \\ &\le \|\boldsymbol{U}_{C,k}\boldsymbol{U}_{C,k}^\top - \boldsymbol{U}_C\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{U}_C^\top\|_F\|\boldsymbol{X}\|_2, \end{aligned} \tag{8}$$

where (8) follows from submultiplicity for Schatten p-norms. Since for any $\boldsymbol{A}$ and $\boldsymbol{B}$, $\|\boldsymbol{A}\|_F^2 = \operatorname{tr}\left[\boldsymbol{A}^\top\boldsymbol{A}\right]$, $\operatorname{tr}[\boldsymbol{AB}] = \operatorname{tr}[\boldsymbol{BA}]$, and $\operatorname{tr}[\boldsymbol{A}+\boldsymbol{B}] = \operatorname{tr}[\boldsymbol{A}] + \operatorname{tr}[\boldsymbol{B}]$,

$$\begin{aligned} &\|\boldsymbol{U}_{C,k}\boldsymbol{U}_{C,k}^\top - \boldsymbol{U}_C\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{U}_C^\top\|_F^2 \\ =\ & \operatorname{tr}\left[\boldsymbol{U}_{C,k}\boldsymbol{U}_{C,k}^\top\right] - \operatorname{tr}\left[\boldsymbol{U}_{C,k}\boldsymbol{Y}_k\boldsymbol{Y}^\top\boldsymbol{U}_C^\top\right] \\ & -\operatorname{tr}\left[\boldsymbol{U}_C\boldsymbol{Y}\boldsymbol{Y}_k^\top\boldsymbol{U}_{C,k}^\top\right] + \operatorname{tr}\left[\boldsymbol{U}_C\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{U}_C^\top\right] \\ =\ & \operatorname{tr}[\boldsymbol{I}_k] - 2\operatorname{tr}\left[\boldsymbol{Y}_k^\top\boldsymbol{Y}_k\right] + \operatorname{tr}\left[\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{Y}\right] \\ =\ & \left\{\operatorname{tr}[\boldsymbol{I}_{k'}] - 2\operatorname{tr}\left[\boldsymbol{Y}^\top\boldsymbol{Y}\right] + \operatorname{tr}\left[\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{Y}\right]\right\} \\ & +(k-k') + 2\operatorname{tr}\left[\boldsymbol{Y}_k^{\perp\top}\boldsymbol{Y}_k^\perp\right] \\ =\ & \|\boldsymbol{I}_{k'} - \boldsymbol{Y}^\top\boldsymbol{Y}\|_F^2 + 2\|\boldsymbol{Y}_k^\perp\|_F^2 + (k-k'). \end{aligned} \tag{9}$$

Remaining derivation for $\|\boldsymbol{I}_{k'} - \boldsymbol{Y}^\top\boldsymbol{Y}\|_F^2$ and $\|\boldsymbol{Y}_k^\perp\|_F^2$ is identical to Lemma 2 of [14], then lemma follows. ■

Lemma 2. *Let $\delta \in (0,1), \eta = \sqrt{\frac{2\log(2/\delta)\alpha(r,m-r)}{r}}$. For $\epsilon > 0$, if $r \ge 1/(\gamma^2\epsilon^2)$, then with probability at least $1-\delta$,*

$$\frac{1}{\gamma\|\boldsymbol{C}\|_2^2}\|\boldsymbol{C}^\top\boldsymbol{C} - \boldsymbol{W}^\top\boldsymbol{W}\|_F \le m\epsilon(1+\eta). \tag{10}$$

Proof: By applying Corollary 1 with δ, we get

$$\|\boldsymbol{C}^\top\boldsymbol{C} - \boldsymbol{W}^\top\boldsymbol{W}\|_F \leq \frac{m(1+\eta)}{\sqrt{r}}\|\boldsymbol{C}\|_2^2. \tag{11}$$

Lemma immediately follows from our choice of r. ■

By combining Theorem 3, Lemma 1 and 2, we get Theorem 2 by assigning $2/\delta$ for each events, Theorem 3 and Lemma 2, and applying the union bound.

VI. Conclusions

In this paper, we proposed a scalable learning to hash algorithm, in which the generalized Nyström method with uniform sampling is applied to PCA + ITQ [8]. With the large-scale high-dimensional dataset, Holidays + Flickr1M [9], we showed that by only using 5% of samples and 30% of features, the method achieves comparable performance to the case of full PCA as well as non-uniform sampling, while the overall computation time is reduced to less than half. We also provided the first theoretical bound analysis of the generalized Nyström method with uniform sampling.

Acknowledgments

This work was supported by NIPA-MSRA Creative IT/SW Research Project, NIPA ITRC Support Program (NIPA-2012-H0301-12-3002), National Research Foundation (NRF) of Korea (2012-0005032), MKE and NIPA ”IT Consilience Creative Program” (C1515-1121-0003), and NRF World Class University Program (R31-10100).

References

[1] A. Gionis, P. Indyk, and R. Motawani, “Similarity search in high dimensions via hashing,” in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 1999.

[2] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 20. MIT Press, 2008.

[3] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching,” *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[4] R. Salakhutdinov and G. Hinton, “Semantic hashing,” in *Proceeding of the SIGIR Workshop on Information Retrieval and Applications of Graphical Models*, 2007.

[5] J. Wang, S. Kumar, and S. F. Chang, “Semi-supervised hashing for scalable image retrieval,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, 2010.

[6] S. Kim and S. Choi, “Semi-supervised discriminant hashing,” in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Vancouver, Canada, 2011.

[7] J. Wang, S. Kumar, and S. F. Chang, “Sequential projection learning for hashing with compact codes,” in *Proceedings of the International Conference on Machine Learning (ICML)*, Haifa, Israel, 2010.

[8] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, 2011.

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, 2010.

[10] A. Frieze, R. Kannan, and S. Vempala, “Fast Monte-Carlo algorithms for finding low-rank approximations,” *Journal of the ACM*, vol. 51, no. 6, pp. 1025–1041, 2004.

[11] C. K. I. Williams and M. Seeger, “Using the Nyström method to speed up kernel machines,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 13. MIT Press, 2001.

[12] S. Kumar, M. Mohri, and A. Talwalkar, “On sampling-based approximate spectral decomposition,” in *Proceedings of the International Conference on Machine Learning (ICML)*, Montreal, Canada, 2009.

[13] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, “A theory of pseudoskeleton approximations,” *Linear Algebra and Its Applications*, vol. 261, pp. 1–21, 1997.

[14] P. Drineas, R. Kannan, and M. W. Mahoney, “Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix,” *SIAM Journal on Computing*, vol. 36, no. 1, pp. 158–183, 2006.

[15] ——, “Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition,” *SIAM Journal on Computing*, vol. 36, no. 1, pp. 184–206, 2006.

[16] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, “Relative-error CUR matrix decompositions,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 844–881, 2008.

[17] S. Kumar, M. Mohri, and A. Talwalkar, “Sampling techniques for the Nyström method,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Clearwater Beach, FL, 2009.

[18] 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*, vol. 6, pp. 2153–2175, 2005.

[19] A. Krizhevsky and G. E. Hinton, “Learning multiple layers of features from tiny images,” Computer Science Department, University of Toronto, Tech. Rep., 2009.

[20] T. S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “NUS-WIDE: a real-world web image database from national university of singapore,” in *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*, Santorini, Greece, 2009.