Efficient Learning for Hashing Proportional Data

Zhao Xu, Kristian Kersting and Christian Bauckhage
Fraunhofer IAIS
Schloss Birlinghoven, Sankt Augustin, 53754, Germany
{*firstname.lastname*}@*iais.fraunhofer.de*

***Abstract*—Spectral hashing (SH) seeks compact binary codes of data points so that Hamming distances between codes correlate with data similarity. Quickly learning such codes typically boils down to principle component analysis (PCA). However, this is only justified for normally distributed data. For proportional data (normalized histograms), this is not the case. Due to the sum-to-unity constraint, features that are as independent as possible will not all be uncorrelated. In this paper, we show that a linear-time transformation efficiently copes with sum-to-unity constraints: first, we select a small number K of diverse data points by maximizing the volume of the simplex spanned by these prototypes; second, we represent each data point by means of its cosine similarities to the K selected prototypes. This maximum volume hashing is sensible since each dimension in the transformed space is likely to follow a von Mises (vM) distribution, and, in very high dimensions, the vM distribution closely resembles a Gaussian distribution. This justifies to employ PCA on the transformed data. Our extensive experiments validate this: maximum volume hashing outperforms spectral hashing and other state of the art techniques.**

***Keywords*-Spectral Hashing; Proportional Data; von Mises Distribution; Dimensionality Reduction**

I. Introduction

Very large datasets with billions of entries are becoming ever more common; Facebook hosts about a billion active users, data collected from Google books or Flickr images can easily exceed billions of terms or pixels. Services like these open up completely new and interesting applications for information retrieval, computer vision and machine learning techniques. For instance, in order to tag an image we are going to upload to Flickr, we could simply search for similar images in Flickr and recommend tags based on annotations of the retrieved images. However, as Weiss *et al.* [34] nicely point out, although conceptually simple, actually applying such ideas requires highly efficient methods for storing millions of items in memory and to perform ultra-fast nearest-neighbor searches. Hashing approaches have emerged as elegant ways to address both these issues ([18], [1], [27], [26], [29], [34]). Intuitively, each item in a dataset is encoded as a compact binary code such that similar items are close to each other in the Hamming space. Retrieving similar neighbors is then done simply by computing the Hamming distances between the binary codes, the number of positions at which the corresponding bits are different. Typical approaches to learning good mappings to Hamming spaces are semantic hashing [26] and spectral hashing [34], where the latter was reported to outperform the former.

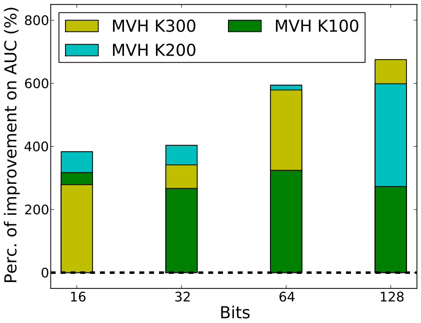

Figure 1. Maximum volume hashing (MVH) as introduced in the present paper can significantly outperform spectral hashing (SH). Shown are the percentage of improvement in the *area under the precision-recall* curve of MVH over SH on the ImageNet dataset consisting of 1,261,406 images. The larger, the better. Positive values indicate that MVH performs better, negative values indicate that it performs worse. (Best viewed in color)

To quickly learn the codes, spectral hashing (SH) is typically based on principle component analysis (PCA), followed by computing the B smallest eigenfunctions of the Laplacian, the sign of which yields the binary code. This is justified only if the dataset is Gaussian distributed. Consider e.g. to hash images or text documents represented as (normalized) histograms of (visual) word occurences as commonly done within information retrieval tasks. Here, the complexity arises from representing the data as vectors that are not only very high dimensional (and often sparse) but also proportional, i.e., each data point describes the relative contributions of K components that must sum to unity. The “sum-to-unity” constraint introduces a bias towards negative values among the correlations when using PCA; variables that are “as independent as possible” will not all be uncorrelated. Consequently, the PCA requires more eigenvectors to explain the variance of the data. Or, in terms of spectral hashing, we have to spend unnecessary bits to achieve good retrieval performance.

Our main contribution is to show that a linear-time transformation can successfully and efficiently deal with sum-to-unity constraints. Using recent data-driven matrix factorization techniques [31] we

1) select a small number K of diverse data points that maximize the volume of the simplex they span using

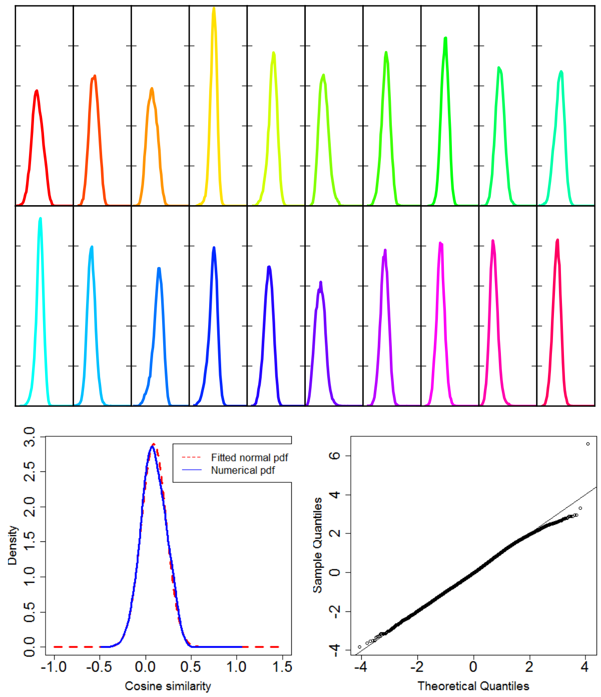

Figure 2. Cosine distances on the LabelMe dataset are essentially Gaussian distributed. (Top) The distributions of cosine similarities between LabelMe images and 20 bases maximizing the volume of the simplex spanned by these prototypes, one curve for each base. Recent results suggest that they are *von Mises* (vM) distributions that arise naturally for proportional data. As one can see, in the high-dimensional case, vM closely resembles a Gaussian distribution. (Bottom) This is validated by a distribution fit (left) and Q-Q plot (right) for the very first dimension selected by MVG. The Q-Q plot compares the empirical distribution on the vertical axis to the Gaussian one on the horizontal axis. (Best viewed in color)

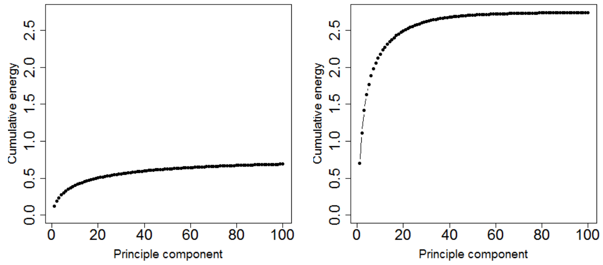

Figure 3. PCA in the transformed space captures the variance in the data better. In turn spectral hashing based on PCA requires less many bits in the transformed space. Shown are the cumulative energies of PCA in the original space (left) and in the transformed space (right).

cosine similarity, and

2) represent each data point by its cosine similarities to the selected ones.

Recent empirical results by Banerjee *et al.* ([3]) suggest that each dimension in the transformed space is likely to follow a *von Mises* (vM) distribution [33]. The vM is a distribution that arises naturally for proportional data and is akin to the Gaussian distribution [23], [3] in particular for large scale, high-dimensional data as also illustrated in Fig. 2. Since it closely resembles a Gaussian distribution, it is justified to employ PCA on the transformed data: PCA requires less many eigenvectors in the transformed space than in the original space, see Fig. 3. Our experiments on several synthetic and real-world datasets validate this intuition. The codes of the tansformed proportional data outperform spectral hashing and other recent and state of the art methods. Fig. 1 illustrates this on 1,26 million images from ImageNet. The running time is O(NK) for selecting K latent factors from N examples, and the time for running SH on N examples with K features, i.e., in total essentially $O(N) + O(SH)$ assuming $K \ll N$. Since SH has to touch at least each example once, asymptotically no overhead is introduced. Moreover, the proposed method is easily parallelizable.

We proceed as follows. We start off by touching upon related work. Then we recap spectral hashing and introduce our transformation based variant, called maximum volume hashing (MVH). Before concluding, we present our experimental evaluation.

II. Related Work

Traditional approaches to nearest-neighbor search partition the data recursively. Examples include kd-trees [15], M-trees [9], and cover trees [5]. Recently, embedding approaches that map data into low dimensional Hamming space have received increasing attention. In contrast to classical dimensionality reduction within Euclidean spaces, the binary embedding supports fast similarity search. Probably the most popular hashing method is locality sensitive hashing ([18], [1]). It hashes data with random linear projection such that the similar data points have higher probability of collision than the dissimilar ones. Salakhutdinov and Hinton ([27], [26]) introduced semantic hashing, which uses a restricted Boltzman machine to compute the codes. Parameter-sensitive hashing [29] converts the hashing problem to a classification problem and solves it using AdaBoost. Weiss *et al.* [34] introduced spectral hashing (SH) and proved its success empirically on a number of real-world datasets. In particular, spectral hashing computes codewords by thresholding a subset of eigenvectors of the similarity graph Laplacian. Many follow-ups on spectral hashing have recently been proposed. For instance, (semi-)supervised variants, see e.g. [19], make use of semantic similarity given in terms of labeled pairs of images. The hash function can be learned in a way correcting the errors made by previous one sequentially. The anchor graph hashing AGH [21] learns compact codes via solving graph Laplacian eigenvectors of an approximate neighborhood graph. The out-of-sample extension is implemented with a Nyström method. In contrast to AGH, we provide a formal justification to use spectral hashing via the connection to von Mises (vM) distribution. In turn the theory underlying SH carries over. Specifically, the selected extreme points maximize the volume of the simplex they span, and the transformed data from the proportional samples reveals a von Mises distribution property in each embedding dimension. Closely resembling Gaussian distributions, the lower-dimensional embedding can smoothly integrate with the analytical eigenfunction formulas to efficiently compute the binary codes according to Weiss et al.'s formulation, instead of using a Nyström method to approximate the eigenfunctions based on the basis points as done in AGH. Moreover AGH employs k-means to get the approximate neighborhood graph and resorts to subsampling to handle gigantic data. This step is of heuristic nature. To the best of our knowledge, hashing proportional data, however, has not received any attention yet. Moreover, since we propose a transformation of the original data, any of the just mentioned techniques can still be applied to the transformed data.

III. Spectral Hashing

Assume that there is a corpus of N documents containing V unique words (vocabulary). The word-document data forms a $V \times N$ matrix X, where $x_{v,i}$ represents the term frequency of word v in document i. With x_i we denote the column vector representing all the words in the i-th document. The codeword of a document is denoted as y_i, which is a B-dimensional binary vector. Vertically stacking all y_is yields Y, the code matrix for all documents with dimensionality $B \times N$.

In general, there are many ways to define a good encoding. Typical criteria include, among others, minimizing reconstruction errors (e.g. principal component analysis), preserving locally linear structures of manifolds (e.g. locally linear embedding [25]), or preserving pairwise distances (e.g. multidimensional scaling [12]). Triggered by the great successes of spectral modeling in many applications, we aim at "preserving the neighborhood relationship of data points" so that Hamming distances between the codes of documents that are neighbors in the input space are smaller than those between codes of non-neighbor documents [4]. Under this premise, the code computation can be formulated as an optimization problem:

$$\begin{aligned} \min_Y & \sum\nolimits_{i,j} w_{i,j} ||y_i - y_j||^2 \\ \text{s.t.: } & y_i \in \{-1, 1\}^B, \\ & \sum\nolimits_i y_i = 0, \\ & \frac{1}{N} \sum\nolimits_i y_i y_i^T = I, \end{aligned} \tag{1}$$

where $w_{i,j}$ represents the similarity between documents i and j which can be defined using any Mercer kernel, e.g. RBF kernels

$$w_{i,j} = \exp(-\frac{||x_i - x_j||^2}{\sigma^2}) \ .$$

The intuition behind this formulation is that the objective function considers distances in the input space *and* in the Hamming space. Consequently, the optimized codes Y^* satisfy that the more similar two documents (large $w_{i,j}$) the smaller their Hamming distance. The centering constraint $\sum_i y_i = 0$ centers coordinates at the origin; the orthogonality constraint $\frac{1}{N} \sum_i y_i y_i^T = I$ forces codes to have unit covariance.

Following [34], we reformulate the optimization problem:

$$\begin{aligned} & \min\nolimits_Y Tr(YLY^T) \\ & \text{s.t.: } y_i \in \{-1, 1\}^B, \\ & \qquad Y^T \mathbf{1} = 0, \\ & \qquad YY^T = I, \end{aligned} \tag{2}$$

where $L = D - W$ is the Laplacian and D is a diagonal matrix with $d_{i,i} = \sum_j w_{i,j}$.

Solving this problem is difficult and several approximate solutions have been proposed. For instance, by relaxing the constraint $y_i \in \{-1, 1\}^B$, the problem turns into a standard eigenvalue problem. The optimized matrix Y^* contains the eigenvectors associated with the B lowest eigenvalues of L [4]. The eigenvectors can be thresholded at zero to obtain binary codes.

However, here our focus is not on the eigenvectors but on the eigenfunctions because they allow us to efficiently compute codes for previously unseen data points. Following [34], we assume that $x_i, x_j \in \mathcal{R}^V$ are i.i.d. drawn from a distribution $p(x)$. Here, $W(x_i, x_j)$ is a weight function to measure similarity between two vectors x_i and x_j. We then seek a projection function $f : \mathcal{R}^V \to \mathcal{R}^B$, which minimizes the expected loss on the data similarity in the Hamming space. That is

$$\begin{aligned} & \min\nolimits_f \int ||f(x_i) - f(x_j)|| W(x_i, x_j) p(x_i) p(x_j) dx_i dx_j \\ & \text{s.t.: } \int f(x) p(x) dx = 0 \\ & \qquad \int f(x) f(x)^T p(x) dx = I. \end{aligned} \tag{3}$$

When the weight $W(x_i, x_j)$ is defined in terms of RBF kernels and x is drawn from a one-dimensional Gaussian distribution[1] $\mathcal{N}(0, \tau^2)$, the solutions to the optimization problem (3) are eigenfunctions Φ_b with eigenvalues λ_b [24], [11]:

$$\lambda_b = (\tau/(\tau + \sigma))^b \tag{4}$$

$$\Phi_b(x) = p_k(x) \exp(-\frac{x^2}{4(\tau + \sigma)}) \tag{5}$$

[1] If x follows a uniform distribution on $[\alpha, \beta]$, then the eigenfunctions and eigenvalues are $\Phi_b(x) = \sin(\pi/2 + b\pi x/(\alpha - \beta))$ and $\lambda_b = 1 - \exp(-\frac{\sigma^2}{2}(\frac{b\pi}{\alpha - \beta})^2)$. [34] report it generates similar result as the Gaussian based solution.

where $p_b(x)$ is a Hermite polynomial of degree b. Using eigenfunctions, documents can be encoded efficiently. If the value of a document i at a bit b is computed with the b'th eigenfunction $\Phi_b(x_i)$, then its code at this bit simply results from thresholding at zero. In the multi-dimensional case, we assume that the data is drawn from a multivariate Gaussian $\mathcal{N}(0, \Sigma)$ with $\Sigma = XX^T$ and determine an orthonormal transformation matrix Λ by PCA that diagonalizes the covariance matrix $C = \hat{X}\hat{X}^T$. In other words, the transformed data $\hat{X}$ are uncorrelated. Therefore we can collect B eigenfunctions for each dimension of $\hat{X}$ and then choose the B smallest ones among the BV eigenfunctions to obtain the multi-dimensional solution.

IV. Maximum Volume Hashing

The derivation of spectral hashing indicates its limitations w.r.t. proportional data. First, the vocabulary (i.e. V) of large corpora can be very large, thus it is generally difficult to compute the transformation Λ directly. Second and more importantly, the data resides on a simplex. Consequently, ℓ^2 norm based weight functions and Gaussian distributions will be no longer suitable. The key idea to overcome these two limitations is to leverage distance geometry in order to relax the assumption of spectral hashing made on the data distribution.

Specifically, we consider a matrix factorization setting where the attribute matrix X is represented as: $X \approx \tilde{X}Z$, where $\tilde{X}$ is a $V \times K$ base matrix ($K \ll V$), and Z is a $K \times N$ factor matrix. Since we consider proportional data, we can restrict ourselves to nonnegative factors [20], [14]. Intuitively, matrix factorization can be interpreted as clustering: $\tilde{X}$ denotes the cluster centroids (bases) and Z contains cluster membership indicators. With the factorization, the optimization problem is represented as:

$$\begin{aligned}
&\min_{g,\tilde{X}} \int ||g(z_i) - g(z_j)||\tilde{W}(z_i, z_j)p(z_i)p(z_j)dz_i dz_j \\
&\text{s.t.:} \int g(z)p(z)dz = 0 \;, \\
&\qquad \int g(z)g(z)^T p(z)dz = I \;, \qquad (6)
\end{aligned}$$

where $\tilde{W}(z_i, z_j) = \exp(-\frac{(z_i - z_j)^T \tilde{C}(z_i - z_j)}{\sigma^2})$ is the transformed weight function $\tilde{C} = (\tilde{X})^T \tilde{X}$.

Unfortunately, it is difficult to analytically solve (6). We therefore propose an approximative solution consisting of two sub-optimization procedures. In the first step, we computes the bases $\tilde{X}$ and the coefficients Z that best explain the attribute matrix X. In the second step, we find the codes Y which best preserve the neighborhood relationship computed with the optimized Z.

How do we select the bases? One attractive way is to directly select them from the data [17]. Corresponding approaches yield naturally interpretable results, since they embed the data in lower dimensional spaces whose basis vectors correspond to actual data points. They are guaranteed to preserve properties such as sparseness or non-negativity and enjoy increasing popularity in the data mining community, see e.g. [16], [30], [22], with important applications to fraud detection, fMRI segmentation, collaborative filtering, and co-clustering.

But which data points are actually "good representatives"? To achieve a low reconstruction error in terms of the Frobenius norm, it was shown that a *good* subset of columns maximize their volume [10]. Specifically, given a matrix X, we select K of its columns such that the volume $Vol(\tilde{X}) = |\det \tilde{X}|$ is maximized, where $\tilde{X}$ contains the selected columns. This criterion, however, is provably NP-hard [10]. A linear-time approximation, called Simplex Volume Maximization was recently introduced by [31] and empirically proven to be quite successful.

Employing well-known results from distance geometry [8], Thurau *et al.* showed that the k-th basis that maximizes locally the simplex volume given the first $k-1$ bases can be found using

$$\begin{aligned}
\phi_{k,i} &= \phi_{k-1,i} + d(\tilde{x}_{k-1}, x_i), \\
\lambda_{k,i} &= \lambda_{k-1,i} + d(\tilde{x}_{k-1}, x_i)^2, \\
\rho_{k,i} &= \rho_{k-1,i} + d(\tilde{x}_{k-1}, x_i) \times \phi_{k-1}, \\
\tilde{x}_k &= \arg\max_i \left[\phi_{k,i} d_{max} + \rho_{k,i} - \frac{k}{2}\lambda_{k,i}\right], \qquad (7)
\end{aligned}$$

where $d(\tilde{x}_{k-1}, x_i)$ denotes the distance between the $k-1$-th basis and the data point i, in our case based on the cosine similarity. After getting all bases $\tilde{X}$, the coefficient matrix Z can be computed straightforward. Since the coefficients of data points i and j are independent, we parallelize the computation and solve the constrained quadratic optimization problem: $\min_{z_i} ||x_i - \tilde{X}z_i||$, $s.t. \sum_k z_{i,k} = 1, z_{i,k} \geq 0$.

What do we gain by this data-driven factorization? Classical matrix factorizations confront us with the difficulty of characterizing the data point distributions in a unified parametric and interpretable form. This is generally intractable. In contrast, when mapping the data points onto a simplex spanned by extreme data points, there are natural parametric distributions. For instance, we can assume that the simplex samples Z follow a Dirichlet distribution, e.g. [7]. However, the Dirichlet components have a special and near-independence structure, it may be too simple to analyze the simplex data where the underlying components are dependent, e.g. [6]. Alternatively, we can assume that z_i follows a logistic-normal distribution $z_i \sim \mathcal{L}(0, \Sigma)$ [2]:

$$|2\pi\Sigma|^{-\frac{1}{2}} \left(\prod_{k=1}^{K} z_{i,k}\right)^{-1} \exp\left[-\frac{1}{2}\log\frac{z_i^T}{z_{i,K}}\Sigma^{-1}\log\frac{z_i}{z_{i,K}}\right],$$

and thus the inverse log ratio transformation

$$\hat{z}_i = \log(z_i / z_{i,K})$$

Algorithm 1: The MVH method for efficiently learning to hash proportional data

```
Input: $X$ (attribute matrix of the training data), $X^*$ (attribute matrix of the test data), $K$ (num. of bases), $B$ (num. of bits)
Training Procedure begin
    Find $K$ extreme data points of $X$ with Eq. (7);
    Compute the transfomed data $Z$ using cosine similarity or Frobenius norm reconstruction based on the $K$ extreme points;
    Find the principal components $P$ of $Z$ with PCA. Each column of $P$ is an eigenvector of $ZZ^T$. Order them by eigenvalues, highest to lowest;
    Along every PCA direction, compute the $B$ smallest eigenvalues with Eq. (4). From the list of $BK$ eigenvalues, find the $B$ smallest ones and the corresponding eigenfunctions $\Phi_b$ based on Eq. (5);
    Threshold the eigenfunction values $\Phi_b(P^T Z)$ at zero and obtain binary codes $Y$ for training data.
Test Procedure (out of sample extension) begin
    Transform the test data $X^*$ to $Z^*$;
    Cast $Z^*$ to the PCA direction with $P^T Z^*$;
    Given the learned eigenfunctions, compute binary codes $Y^*$ by thresholding $\Phi_b(P^T Z^*)$ at zero.
Output: $Y$, $Y^*$ (code matrix for training and test data), principal components $P$, and a list of eigenfunctions $\Phi_b$
```

follows a Gaussian distribution $\hat{z}_i \sim \mathcal{N}(0, \Sigma)$. Applying this transformation to the learned coefficient matrix Z, the optimization problem is simplified as

$$\min_f \int ||f(\hat{z}_i) - f(\hat{z}_j)|| W(\hat{z}_i, \hat{z}_j) p(\hat{z}_i) p(\hat{z}_j) d\hat{z}_i d\hat{z}_j$$

$$\text{subject to: } \int f(\hat{z}) p(\hat{z}) d\hat{z} = 0$$

$$\int f(\hat{z}) f(\hat{z})^T p(\hat{z}) d\hat{z} = I, \qquad (8)$$

with eigenfunctions ϕ_b and eigenvalues λ_b equivalent to (5). Thus, we can again compute the multivariate solution by finding the orthonormal matrix of $\hat{Z}$ with PCA. Note that the result will be back-transformed to the simplex with $z_i = \exp(\hat{z}_i)/(1 + \sum_k \exp(\hat{z}_{i,k}))$. The complete procedure of the MVH method is summarized in Alg. 1.

However, we can actually do considerably better. Recent empirical results by [3] suggest that proportional data is likely to follow a *von Mises* (vM) distribution [33]. It is a distribution that arises naturally for proportional data and is akin to the Gaussian distribution [23]. Since we have selected real data points as basis vectors or cluster centers, this still holds for the low-dimensional embedding. Consequently, we can simply compute the reconstruction coefficients as the cosine similarities between the data points and the cluster centers. This is what we compute anyhow when selecting the cluster centers. Since the vM distribution closely resembles a Gaussian distribution, this justifies using PCA on the transformed data.

V. Experimental Evaluation

Our main intention was to investigate the following question

(Q) "Can maximum volume hashing (MVH) improve upon the retrieval performance of spectral hashing (SH) for proportional data?".

To this end, we implemented MVH in Python and evaluated the performance of MVH for nearest neighbor retrieval on synthetic and real-world datasets. We compared MVH to SH and Zhang *et al*'s [35] SVM-based hashing method[2]. For the MVH method, we investigated two strategies for computing the coefficients Z:

1) cosine similarity (denoted as CSC) for proportional data, and
2) Frobenius norm reconstruction (denoted as MRC) for arbitrary data.

In both cases, the cosine similarity was used to sub-select columns. We report on the area under the precision-recall curve of the retrieval results.

A. The Non-Proportional Data Case

The key assumption of SH is that data points lie in a Euclidean space. This assumption is not always valid and poses a challenge for several retrieval applications where data commonly lies in complex non-Euclidean manifolds. However, although not the main target of the present paper, SiVM actually turns any dataset into proporational data as long as we have an appropriate distance at hand. To see whether doing so can be beneficial, we investigated the following sub-question

(Q1) "Can MVH preserve the semantic similarities of data points drawn from different distributions and spaces even if they are not proportional?"

To this aim, we considered three situations, namely (1) a three dimensional manifold (Swiss roll), (2) a Dirichlet distribution, and (3) mixtures of Gaussians in the 2D Euclidean space. The experimental results are shown in Fig.4, where K denotes the number of bases in matrix factorization. As one can see, on manifold data and simplex data, the MVH method outperformed SH. As expected, for the Gaussian data, the MVH did not outperform SH. This is because the similarity between samples in the dataset can be well represented using ℓ^2 norm based distance functions and the samples in the dataset are drawn from Gaussians. Thus, the

[2] Here the training labels are "learned" with spectral hashing method. Alternatively one can employ, for example, locally linear embedding (LLE), multidimensional scaling (MDS), Isomap etc. to get pseudo-labels.

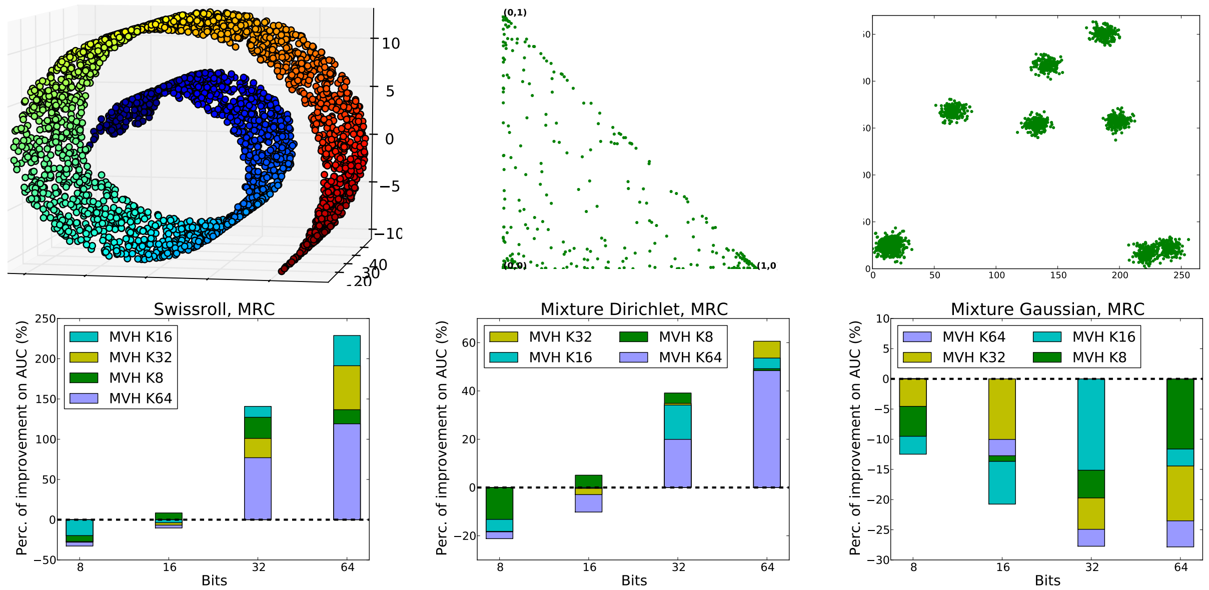

Figure 4. Experiment results on Swiss roll (left), mixture Dirichlet (middle) and mixture Gaussian (right) data. The coefficients Z in the MVH are computed with Frobenius norm reconstruction. (Best viewed in color)

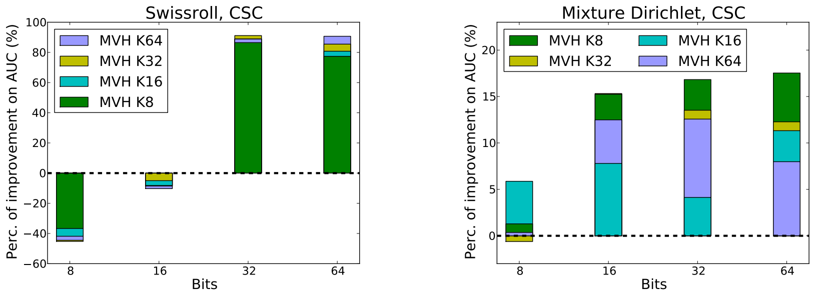

Figure 5. Experiment results of the MVH on the Swiss roll data (left) and the mixture Dirichlet data (right) with the cosine similarity coefficients. (Best viewed in color)

characteristics of the data meet the assumptions of SH but the approximate matrix factorization employed within MVH might introduce some loss. Fig. 5 shows the performance of MVH using the cosine similarity coefficients. It shows a similar tendency as above. Note that, we do not evaluated this variant on the Gaussian data, since cosine similarities cannot reasonably applied to Gaussian samples. To summarize, the first experimental results demonstrate that, for data with complex distribution, MVH can indeed preserve the semantic similarities in the mapped spaces. This is clearly an affirmative answer to question **(Q1)**.

B. The Proportional Data Case

In our main set of experiments, we focused on proportional data. Sepcifically, we investigated

(Q2) "Can MVH encode the neighborhood relationship of proportional data in the Hamming space better than SH?"

(Q3) "Does MVH scale well to large datasets?"

(Q4) "Does MVH perform well on medium scale datasets?"

To this end, we considered two real-world text and three real-world image datasets, namely Cora and 20newsgroup and Labelme, Peekaboomm and ImageNet, respectively.

1) Large Scale Datasets: The LabelMe and the Peekaboom datasets[3] [32] are collections of images represented with 512-dimensional Gist descriptors. After preprocessing (mean-centering and unit-length normalization), each dataset is randomly split into training (20,019/55,637 images for LabelMe/Peekaboom) and test (2000 images) data. The ground truth neighbors (100 images) were defined by the ℓ^2 distances over the preprocessed Gist vectors.

The ImageNet dataset [13] is a large-scale ontology of images built upon the WordNet structure. The dataset consists of 1,261,406 images with a visual vocabulary of 1000 visual words. We randomly select 2000 images as test data,

[3] http://cs.nyu.edu/~fergus/research/tfw_cvpr08_code.zip

and the rest ones as training data. The ground truth neighbors (100 images) was defined by the cosine similarities over the visual word frequencies.

We analyze the distributions of cosine similarity coefficients on the image datasets. The results on LabelMe data are shown as Fig. 2. As one can see, the numerical distribution of the cosine similarity coefficients is very close to the fitted Gaussian. The Q-Q plot specifies the quantiles of the observed coefficients against the theoritically Gaussian ones, and further illustrates the similarity of the two distributions. The Gaussian-akin distribution property of the transformed data allows for PCA requiring less eigenvectors to explain the variance of the data (see Fig. 3), and in turn spectral hashing based on PCA requires less many bits in the transformed space, no matter how the data is distributed originally.

The experimental results on the three image datasets are summarized in Fig. 6 and Fig. 1. We note that, in all settings (different dimensional hashing codes), MVH outperforms SH. This is clearly an affirmative answer to question **(Q3)**

We illustrate some exemplary retrieval results on the image datasets in Fig. 7 and Fig. 8. Note the small but important differences in the retrieved nearest neighbors.

2) Medium Scale Datasets: For smaller datasets, the assumption that each selected dimension follows a vM distribution that closely resembles a Gaussian distribution might be violated. To investigate how much this impacts the performance of MVH we consideres several medium scale text corpora. The Cora dataset [28] consists of scientific publications. Each publication is represented as a vector whose elements specify the frequency of a word in the publication. The dataset includes 2708 publications with a dictionary of 1433 unique words. We randomly select 1500 documents for training and the remaining 1208 documents for testing. The 20newsgroup dataset[4] consists of newsgroup postings. Each document is represented as a 1000-dimensional vector of term frequencies. After removing documents with less than 10 words, we obtain 16899 documents. They are randomly split into training (15000) and test (1899) sets. The ground truth neighbors (100 documents) in the two datasets were defined by the cosine similarities over the term frequencies.

The cosine similarity coefficients of the two medium scale text datasets are analyzed in Fig. 9. We randomly select five dimensions (bases) to visualize. Even if the empirical distributions are not close to the Gaussian (but similar to mixture of Gaussian), PCA in the transformed space still requires less eigenvectors to explain the variance of the data (see Fig. 10), and in turn spectral hashing based on PCA would require less many bits to encode the neighborhood relationship in the data.

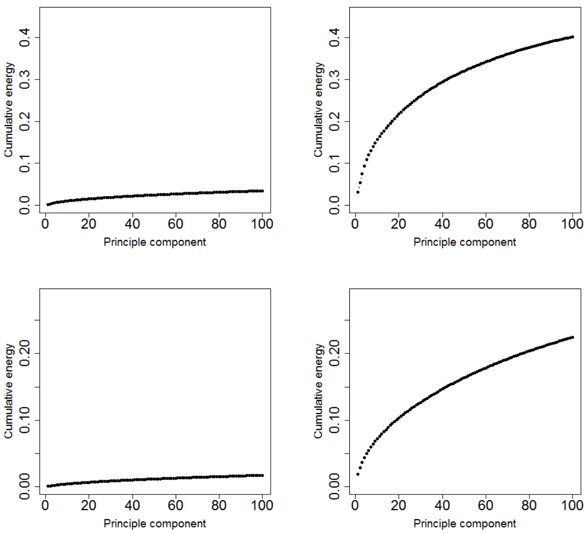

Figure 10. Cumulative energies of PCA for the medium scale text corpora: (top) Cora and (bottom) 20newsgroup datasets. Shown are the results on the original space (left) and the transformed space (right).

The experimental results on the text datasets are summarized in Fig. 11. Again, we find similar tendencies as for large scale image datasets. In all settings (different dimensional hashing codes), MVH outperforms SH. Moreover, the cosine similarity based reconstruction coefficients outperform the ones based on Frobenius norm reconstruction. This stood to be expected since from information retrieval it is known that the cosine similarity can capture similarities between documents more effectively. This is clearly an affirmative answer to question **(Q4)**. Taking the results on **(Q3)** and **(Q4)** together also clearly indicate that question **(Q2)** can be answered affirmatively.

To summarize, all our experiments indicate that codes computed using MVH indeed outperform SH codes for proportional data. Thus, our guiding question **(Q)** can be answered affirmatively.

VI. Conclusion

We have developed a novel efficient spectral hashing approach specifically tailored towards proportional data, called maximum volume hashing (MVH). Due to the “sum-to-unity” constraint of proportional data, features that are “as independent as possible” will not all be uncorrelated. Consequently, spectral hashing based on PCA has to spend unnecessary bits to achieve good retrieval performance. In contrast, MVH exploits distance geometry to select a small number of representative data points in linear-time using the cosine similarity. Since the selected data points maximize the volume of the simplex they span, the transformed data is still likely to follow a von Mises (vM) distribution in each embedding dimension. More importantly, it closely resembles a Gaussian distribution so that we can again employ PCA for efficiently computing binary codes. Our experimental results

[4] http://cs.nyu.edu/˜roweis/data.html

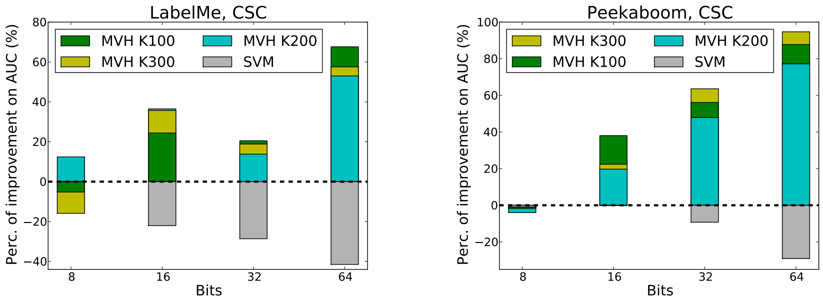

Figure 6. Experiment results on the LabelMe (left) and the Peekaboom (right) datasets. (Best viewed in color)

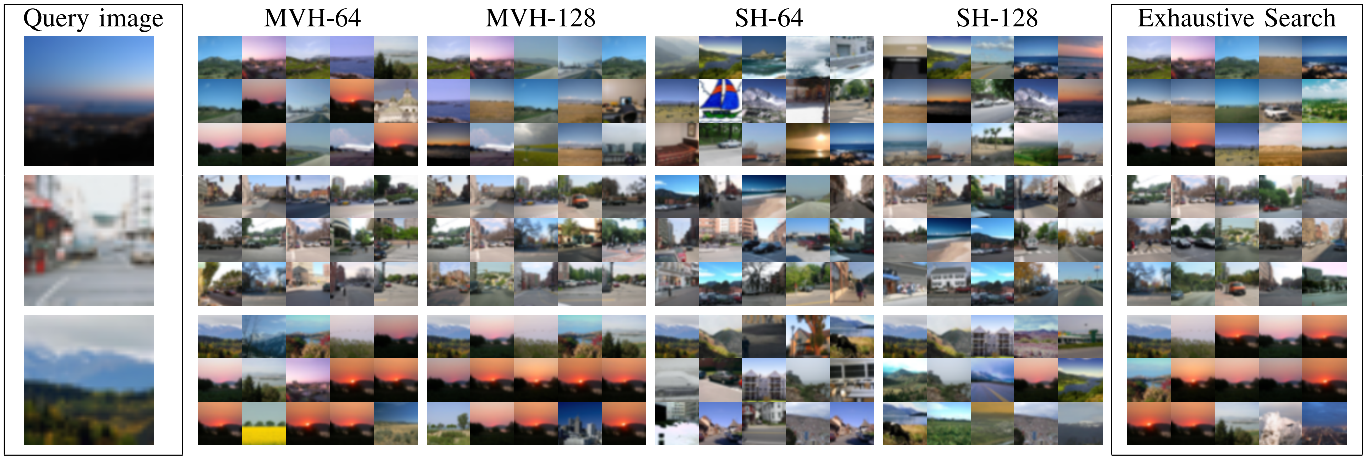

Figure 7. Each row shows the query image and the 15 retrieved nearest neighbors in the LabelMe dataset using, (a) 64-bits MVH, (b) 128-bits MVH, (c) 64-bits SH, (d) 128-bits SH, and (e) exhaustive search based on ℓ^2 distances over Gist descriptors. (Best viewed in color)

validate this. The codes of the transformed proportional data outperform state of the art methods.

Our work provides several interesting avenues for future work. First, instead of running PCA one could employ mixtures of vM resp. their high-dimensional counterpart von Mises Neumann (vMN) distributions [3] for computing codes. Second, one should exploit the straightforward semi-supervised extensions of MVH within retrieval applications. Ultimately, one should find other distances-distributions pairs. For instance, what if the data are vertices in a graph? Indeed we can use random-walk based distances to hash the nodes in the graph. But which distribution to use? Are simplex distributions the only option?

Acknowledgments

This work was funded by the Fraunhofer ATTRACT Fellowship ”Statistical Relational Activity Mining” (STREAM), and by the European Commission under contract number FP7-248258-First-MM.

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[2] J. Atchison and S.M. Shen. Logistic-normal distributions:some properties and uses. *Biometrikae*, 67(2):261–272, 1980.

[3] A. Banerjee, I.S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, pages 585–591, 2001.

[5] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proc. ICML*, 2006.

[6] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, 2006.

[7] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[8] L. M. Blumenthal. *Theory and Applications of Distance Geometry*. Oxford University Press, 1953.

[9] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. VLDB*, 1997.

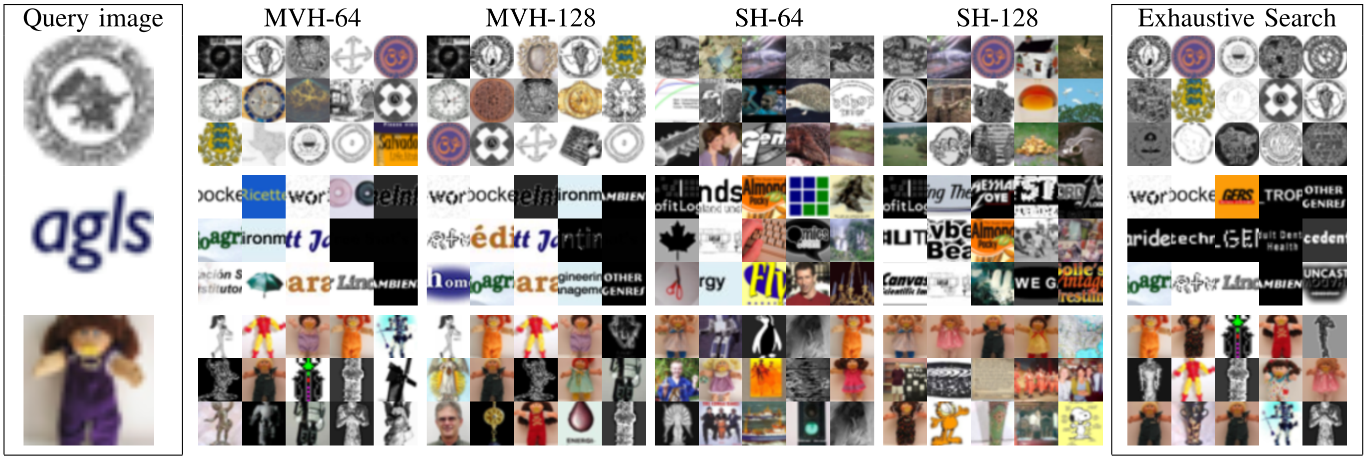

Figure 8. Each row shows the query image and the 15 retrieved nearest neighbors in the Peekaboom dataset using, (a) 64-bits MVH, (b) 128-bits MVH, (c) 64-bits SH, (d) 128-bits SH, and (e) exhaustive search based on ℓ^2 distances over Gist descriptors. (Best viewed in color)

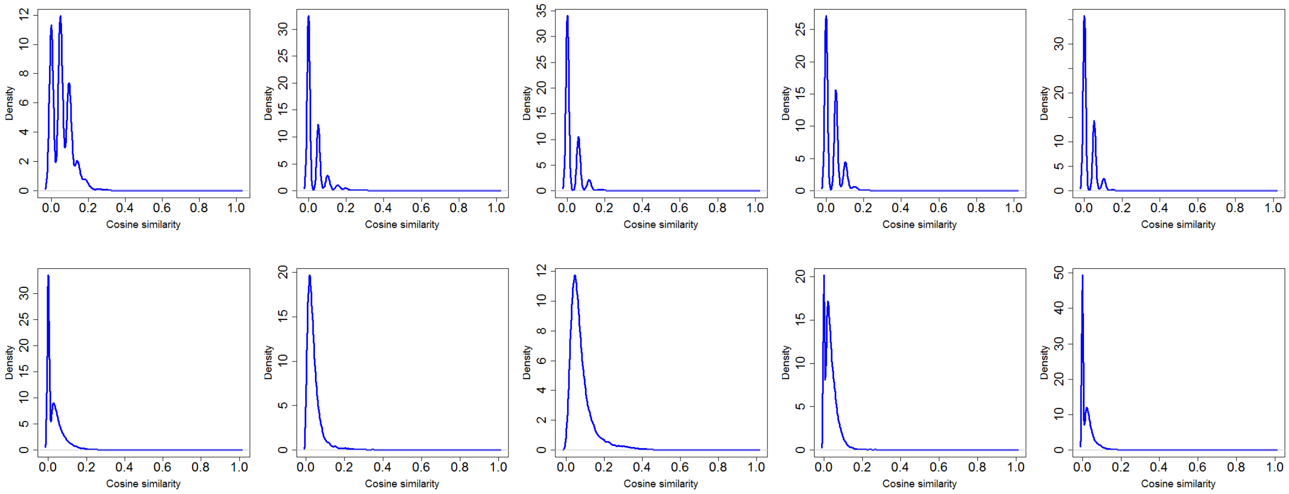

Figure 9. Distributions of cosine similarity coefficients on medium scale text corpora: (top) Cora and (bottom) 20newsgroup datasets.

[10] Ali Civril and M. Magdon-Ismail. On Selecting A Maximum Volume Sub-matrix of a Matrix and Related Problems. *Theoretical Computer Science*, 410(47–49):4801–4811, 2009.

[11] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, F. Warner, and S. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *Proc. of NAS*, volume 102, pages 7426–7431, 2005.

[12] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, London, 1984.

[13] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*, 2009.

[14] C.H.Q. Ding, T. Li, and M.I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on PAMI*, 32:45–55, 2010.

[15] J. Freidman, J. Bentley, and A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

[16] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding lowrank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[17] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1-3):1 – 21, 1997.

[18] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, 1998.

[19] S. Kumar J. Wang and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 2010.

[20] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, 2001.

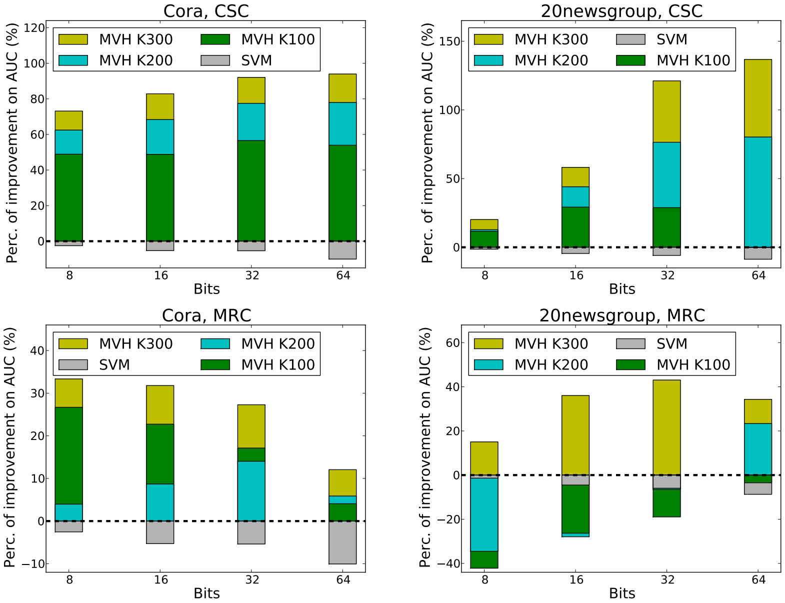

Figure 11. Experiment results on the Cora (left) and 20newsgroup (right) datasets. (Best viewed in color)

[21] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *ICML*, pages 1–8, 2011.

[22] M.W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *PNAS*, 106(3):697–702, 2009.

[23] K.V. Mardia and P. Jupp. *Directional Statistics*. John Wiley and Sons Ltd., 2000.

[24] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[25] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[26] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50:969–978, 2009.

[27] R. Salakhutdinov and G.E. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. *JMLR*, 2:412–419, 2007.

[28] P. Sen, G.M. Namata, M.B., L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[29] G. Shakhnarovich, P.A. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. ICCV*, pages 750–759. IEEE Computer Society, 2003.

[30] Jimeng Sun, Yinglian Xie, Hui Zhang, and Christos Faloutsos. Less is More: Compact Matrix Decomposition for Large Sparse Graphs. In *SDM*, 2007.

[31] C. Thurau, K. Kersting, M. Wahabzada, and C. Bauckhage. Descriptive matrix factorization for sustainability: Adopting the principle of opposites. *Journal of Data Mining and Knowledge Discovery*, 24(2):325—354, 2012.

[32] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proc. CVPR*, pages 1–8. IEEE Computer Society, 2008.

[33] R. von Mises. Über die ”Ganzahligkeit” der Atomgewichte und verwandte Fragen. *Phys. Z.*, 19:190–500, 1918.

[34] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760, 2009.

[35] D. Zhang, J. Wang, D. Cai, and J. Lu. Self-taught hashing for fast similarity search. In *Proc. of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 18–25, 2010.