Local Spectral Analysis and its Applications

Yosi Keller

Email: yosi.keller@yale.edu

Department of Mathematics, Yale University, Connecticut, USA

Abstract

This work presents an approach to local spectral analysis of data, which allows the introduction of prior knowledge into spectral clustering. Our approach is based on propagating graph based diffusions, formulated using the pairwise similarities within the input dataset. A cluster can be identified by detecting discontinuities in the diffusion initiated in a small subset of samples in the cluster of interest. We rigorously analyze this property using the Fokker-Planck interpretation of spectral embeddings, and demonstrate the applicability of our approach by applying it to image segmentation, documents classification and dynamic search in visual databases.

1 Introduction

The analysis of high-dimensional datasets is at the heart of contemporary research, as such information sources are found in a variety of applications, such as web documents search, data mining and multi-spectral imaging to name a few. First introduced in the context of manifold learning and graph partitioning, spectral techniques [1, 2, 3, 4, 5, 6, 7] provide an elegant mathematical framework for non-linear embeddings. Given a set $\Omega = \{x_0, \dots, x_L\}$ of high dimensional points, such schemes start by applying a positive semi-definite kernel W to the set Ω. $\{\psi_i(x)\}$, the set of eigenvectors of W were shown to provide an embedding of Ω into a metric space [2]. More accurately, computing the L_2 metric $\|\psi(x_i) - \psi(x_j)\|$ in the embedding space is equivalent to computing a certain metric $\|x_i, x_j\|$ in the original space Ω. The embedding is then given by

$$\Psi_t : x \longmapsto (\psi_1(x), \psi_2(x), \dots, \psi_L(x))^T \ . \tag{1.1}$$

The choice of the embedding kernel W determines the type of the distance computed $\|x_i, x_j\|$. For instance, using the Normalized graph Laplacian as a kernel results in computing the *Diffusion distance* between the points Ω [6, 7]. In particular, as one can use just a subset of the eigenvectors $\{\psi_i\}$ in Eq. 1.1, this procedure yields a low dimensional representation of the set Ω.

In addition, Belkin and Niyogi [2] explain that, in the case of a dataset approximately lying on a submanifold, this choice corresponds to approximating the heat kernel on the submanifold. A different interpretation to spectral embedding was given by Lafon and Nadler [8], who showed that the spectral eigenvectors identify asymptotically with the eigenvectors of the Fokker-Planck operator on Ω.

It is custom to use the kernel $w\left(x_i, x_j\right)$=$\exp(-\|x_i - x_j\|^2/\varepsilon)$, where $\|x_i - x_j\|$ is an application specific metric that characterizes the local geometry of the set Ω. This corresponds to the notion that the only relevant information available for high dimensional data analysis is related to local distance measurements, and that the global structure of a set should be derived by agglomerating locally defined measures. $\varepsilon > 0$ is a scale parameter defining the extent of the locality of the data points.

The use of spectral embeddings for clustering is related to finding bottlenecks and clusters in graphs [9, 10]. This was first introduced as a relaxation of the discrete optimization problem of finding an optimal cut in a graph [11, 12], where the first non-constant eigenfunction is used as a classification function that partitions the data into two clusters. A random walk view of spectral embeddings was suggested by Meila and Shi [13], and used to extend the spectral approach to finding multiple clusters using multiple eigenvectors. The analysis in [13] relates to lumpable Markov chains and their piecewise constant eigenvectors. A generalization to the non-lumpable case was presented by Lafon and Lee in [14].

Such schemes are mostly applied in unsupervised frameworks, where the entire dataset is analyzed and partitioned to either two [15], or an a-priory given number of clusters [16]. Recently the problem of bottom-top, biased clustering has received particular attention [17, 18], as it arises naturally in many practical situations. Consider the problem of finding a particular cluster $\widehat{\Omega}$ in a large dataset Ω. Using a top-bottom approach, such as [15] or [16] requires the analysis of the entire dataset without any assurance that any of the resulting clusters will correspond to $\widehat{\Omega}$. For instance, consider a dataset represented by a lumpable transition matrix as in [13]. Thus, for a dataset with N clusters $\{C_i\}$, one gets N piecewise constant eigenvectors and the magnitude of the eigenvalue λ_i corresponds to the cardinality of the sets $\{C_i\}$. Hence, in order to identify the $K'th$ cluster, one has to take into account at least K eigenvectors, and for a large K, λ_k might be quite small and C_k difficult to detect. In practice, real datasets are rarely represented by lumpable Markov chains, thus, they lack the piecewise constant eigenvectors, making the detection of C_k difficult. Moreover, the numerical computation of eigenvectors related to small eigenvalues is difficult and might prove to be inaccurate.

A constrained top–bottom approach was presented in [17], where two subsets of a-priori known samples, $\Omega_0^+ \in \widehat{\Omega}$ and $\Omega_0^- \notin \widehat{\Omega}$ were used to provide constraints on a global, two-way, min-cut partitioning, and the scheme requires the computation of the eigenvectors of a $|\Omega| \times |\Omega|$ matrix, where $|\Omega|$ is the cardinality of the set Ω.

Computing the eigenvectors of a Markov chain corresponds to analyzing its asymptotic, where the eigenvectors encode global information over the entire dataset. Thus, in order to analyze and partition only a particular cluster, we turn our attention to the non-asymptotic regime. Such a scheme was studied by Slonim and Tishbi in [19], where a Markov random walk is propagated, based on the data similarity matrix, and initiated by inducing a random distribution vector $P(t_0)$ over all of the samples. At each iteration n they compute the Mutual Information (MI) between $P(t_0)$ and $P(t_n)$. It was experimentally shown that at the point of maximal information loss $\left(\underset{t}{max} \left\{ \frac{\partial MI(P(t_0),P(t_n))}{\partial t} \right\} \right)$, a pseudo-stable state of $P(t_n)$ is achieved and the different clusters can be identified.

A partially labeled set of samples was used by Szummer and Jaakkola in [20] to initiate a random walk over the dataset. The random walk probabilities were then used to classify the unlabeled data based on a probabilistic formulation using either an EM or maximal margin classifiers.

A graph theoretic approach to unsupervised data analysis was suggested by Harel and Koren in [21, 22]. They compute the affinity matrix related to the dataset, and row normalize it to form a Markov matrix. A random walk is then initiated at all of the datapoints, and the probabilities induced by it are used as a set of features defined on the dataset. A new graph is defined based on the inner similarities of this new set of features (probabilities). This new graph is then clustered by a *Clustering by Separation* approach, where a greedy algorithm iteratively removes edges until the clusters are formed.

In this work we present a bottom-top approach to local spectral analysis (LSA). We aim to cluster a subset of Ω denoted $\widehat{\Omega}$. For that we utilize an a priori given subset of $\widehat{\Omega}$ denoted Ω_0. This setup is illustrated in Fig. 1, where Ω is the entire image (Fig. 1a), $\widehat{\Omega}$ is the cluster we aim to segment (the green overlay in Fig. 1b) and Ω_0 is a set of points given within the target cluster (The X mark in Fig. 1b).

By diffusing from the set Ω_0, we are able to analyze the dataset Ω in the vicinity of $\widehat{\Omega}$. In particular, for structured datasets such as images, where each sample is connected to a limited number of other samples, the LSA can be applied on the fly, without having to access the entire dataset at once. For instance, in Fig. 1, when trying to cluster the neck area, there is no need to embed the person's legs area or even compute

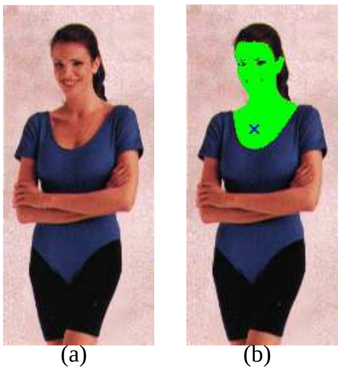


Figure 1: An illustration of Local Spectral Analysis (LSA). We aim to cluster the face and neck area of the woman (the set $\widehat{\Omega}$), using a single sample marked by the X in (b) (the set Ω_0). We only have to analyze the data in the vicinity of $\widehat{\Omega}$ and not the entire image (the set Ω). The clustered area is depicted in (b) by the green overlay.

the corresponding terms of the Markov matrix. In contrast to the previously cited works [19, 20, 21, 22] our approach is based on *diffusion propagation* rather than *probability propagation*. Let M be a Markov matrix, then $p_{n+1}^T = (p_n)^T \cdot M$ defines a Markov random walk, while we propagate the diffusion values $\phi_{n+1} = M \cdot \phi_n$, and ϕ_n is not a probability. By propagating the diffusions we take advantage of their discontinuities across cluster boundaries. In general, we are unable to restrict the diffusion propagation just to $\widehat{\Omega}$, and the discontinuities are used to detect $\partial\,\widehat{\Omega}$, the boundary of the cluster of interest $\widehat{\Omega}$.

We also show that the proposed scheme can provide reliable partitioning even when the initial dataset Ω_0 contains errornous elements that do not belong to $\widehat{\Omega}$. Last, we present an unsupervised approach to partitioning a dataset into K clusters using the local analysis and denote it K-LSA. Our rigorous analysis of the diffusion discontinuities is based on the Fokker-Planck interpretation of spectral embedding [8, 23]. For experimental verification we applied the proposed scheme to image clustering, documents search and dynamic database search. To summarize, we offer the following contributions:

- The LSA semi-supervised partitioning scheme, based on diffusion propagation, where the diffusion is propagated from a given set of samples denoted as seeds.

- A computational approach to identifying the boundary of the target cluster, that is able to handle errornous seeds.

- The K-LSA unsupervised approach to partitioning a dataset into K clusters based on a local spectral analysis.

The paper is organized as follows: we start by introducing the local spectral approach and its rigorous analysis in Section 2. Experimental results are presented in Section 3, while conclusions and future research are discussed in Section 4.

2 Local spectral analysis

The local spectral analysis approach (LSA) is based on the observation that one can consider spectral embedding as a similarity propagation process, where similarities are propagated from a set of known samples $\Omega_0 \in \widehat{\Omega}$, denoted as *seeds*. The common approach to rigorous analysis of spectral data embedding and partitioning, is by considering the properties of the eigenvectors related to random walks [13, 16] or kernels [2, 6, 14]. In contrast, the LSA does not require eigenvectors computation, but utilizes a dynamic process resembling an initial value partial differential equation, based on the locations of the seeds. We formulate the process as a an initial value Fokker-Planck (FP) problem:

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot \left(\nabla \phi + \phi \nabla U\left(x\right)\right), -\infty < x < \infty, \ \ t > 0 \tag{2.1}$$

$$\phi\left(x, x_0, t\right) = \delta\left(x - x_0\right) \text{ as } t \to 0, \tag{2.2}$$

where ϕ is the diffusion function, U is the energy potential and x_0 is the location of a particle/seed. This interpretation of spectral embedding was first suggested by Lafon and Nadler [8, 23] and was originally studied in the context of the numerical analysis of chemical reactions [24, 25]. There, the diffusion represents a molecule trying to overcome a potential barrier as illustrated in Fig. 2.

$\phi\left(x, t\right)$, the solution of Eq. 2.1 is then related to the motion of particle in a potential well, initiated at the location x_1. x_1 and x_3 correspond to stable states, where $\phi\left(x_1\right) \gg \phi\left(x_3\right)$ due to the initiation at x_1, and x_2 is an unstable transition state.

Our data analysis approach is based on two steps:

1. Propagating the diffusion values $\phi\left(x_i\right)$ from the set of seeds Ω_0, x_1 in Fig. 2.

2. Identifying the cluster boundary by detecting the discontinuity in $\phi\left(x_i\right)$, and utilizing the fact that $\phi\left(x_1\right) > \phi\left(x_3\right)$.

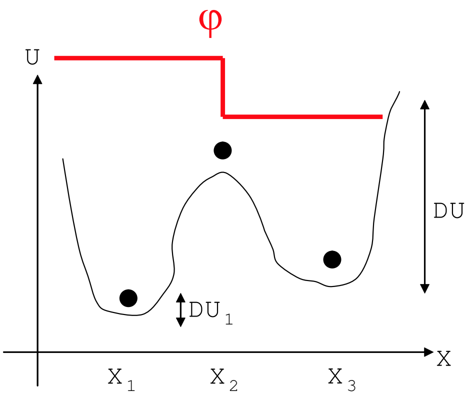


Figure 2: A particle moving in a 1D potential field $U(x)$. If the particle's motion is initiated at x_0, its diffusion will be higher within the potential well surrounding x_1. ϕ, the diffusion function, will be a discontinuous at x_2, the boundary of the cluster around x_1.

More accurately, given the dataset $\Omega = \{x_0, ..., x_L\}$ and the set of seeds Ω_0, we start by forming the Markov matrix $M_{L\times L}$ corresponding to the dataset Ω: following the spectral formulation [6, 7], we choose an application specific metric $\|x_i, x_j\|$ and a kernel W and compute the random walk matrix M

$$m(x_i, x_j) = \frac{w(x_i, x_j)}{\sum_{j=1}^{L} w(x_i, x_j)}.$$

For instance, for images one can use the L_2 norm for pixels or the output of directional filters. A typical choice for W is the Gaussian kernel. Starting with a diffusion vector ϕ_0 that is zero outside Ω_o and one on Ω_o, we propagate ϕ_n using Algorithm 1. The thresholding in Step #5 reduces the leakage of diffusion to very weakly connected samples ($T_0 = O\left(10^{-5}\right)$).

Note that for structured datasets, each sample x_i only interacts with a small number of other samples. For instance, in image segmentation, it is common to restrict the interaction of each pixel to a support of 5×5 or 7×7 pixels, resulting in a sparse Markov matrix M. Thus, there is no need to compute the entire matrix M, whose storage might prove prohibitive and the required sub-matrix of M can be computed on the fly.

Next we show that cluster boundaries are characterized by discontinuities in ϕ_n. For that we consider the solution of Eq. 2.1 given by Schuss in [25] for a single particle in the potential field $U(x)$

$$\phi\left(x,t\right) = \psi_0\left(x\right) + \sum_{n=1}^{\infty} \psi_n\left(x\right)\psi_n\left(x_0\right)e^{-\lambda_n t} \tag{2.3}$$

Algorithm 1 Computing the random walk probability ϕ_n

```
Let $\phi_0$ be the indicator vector that is zero outside $\Omega_o$, and 1 on $\Omega_o$. $n_{\max}$ is a predefined number of time steps.
for $n = 1$ to $n_{max}$ do
    Compute $\widetilde{\phi}_n = M \cdot \phi_{n-1}$
    Define $\phi_n$ by $\phi_n(x) = \widetilde{\phi}_n(x)$ if $x \notin \Omega_0$ and $\phi_n(x) = 1$ otherwise.
    Threshold the 'leakage' diffusions: $\phi_n(x_i) = 0 \; \forall x_i$ s.t. $\phi_n(x_i) < T_0$.
end for
Return $\phi_n$ where $n = n_{\max}$.
```

where $\psi_0(x)$ is the eigenfunction corresponding to the first constant eigenvalue $\lambda_0 = 0$. Applying the initial condition in Eq. 2.2 and choosing a time $t \gg 0$, Eq. 2.3 is reduced to

$$\phi(x,t) = \psi_0(x) + \psi_1(x)\,\psi_1(x_0)\,e^{-\lambda_j t}. \tag{2.4}$$

$\psi_1(x)$ was shown by Schuss et. al. [24, 25] to be discontinuous across $\partial\widehat{\Omega}$, the boundary of $\widehat{\Omega}$ (the point x_1 in Fig. 2). For instance, consider the setup depicted in Fig. 2 with potentials $\Delta U \gg \Delta U_1$. This corresponds to a particle going through quantum states, resulting in a piecewise constant $\phi(x,t)$ that is discontinuous at x_2. An equivalent probabilistic result was derived by Meila and Shi [13] for the eigenvectors of matrices corresponding to lumpable Markov chains. In particular, these piecewise constant eigenvectors are also discontinuous around x_2. In general, datasets are not characterized by lumpable Markov matrices, making the clustering more difficult. Yet, the discontinuities in $\phi(x,t)$ remain (while not being as sharp as a picewise discontinuity) and are useful for clustering.

Moreover, since we initiate the diffusion on the samples $\Omega_0 \in \widehat{\Omega}$, we have that $\phi\left(\widehat{\Omega}\right) > \phi\left(\Omega - \widehat{\Omega}\right)$. Our partitioning scheme utilizes both properties, and computes a threshold value T^* such that any sample, whose diffusion value is larger than T^*, is classified as belonging to $\widehat{\Omega}$. A possible choice for T^* is the diffusion value at the largest discontinuity of ϕ, denoted $T_{\max}$. Yet, for most datasets, the boundary $\partial\widehat{\Omega}$ has a certain width (in terms of ϕ values), thus, we set T^* to be larger than $T_{\max}$. In terms of Fig. 2, this corresponds to setting $T^* = \phi(x_1)$ rather than $T^* = \phi(x_2)$.

More accurately, let $N(T) = |\phi_n > T|$ be the number of samples with diffusion values larger than T, then in order to identify the discontinuity $T_{\max}$, we utilize the observation that $N(T)$ is a monotonically decreasing function of T. $T_{\max}$ is then given by $T_{\max} = \arg\max\limits_{\widetilde{T}} \left|\frac{\partial N(\widetilde{T})}{\partial T}\right|$ and we set T^* to be the first saddle point of $\left|\frac{\partial N(T)}{\partial T}\right|$, such that $T^* > T_{\max}$. This procedure is summarized in Algorithm 2.

Algorithm 2 Computing the threshold T^* and the clustering of $\widehat{\Omega}$

```
1: Apply Algorithm 1 and compute $\phi_n$.
2: Compute $N(T) = |\phi_n(x_i) > T|$ for $T \in [0,1]$, and
       $$T_{\max} = \arg\max_{\widetilde{T}} \left|\frac{\partial N(\widetilde{T})}{\partial T}\right|.$$
3: Set $T^*$ to be the first saddle point of $\left|\frac{\partial N(T)}{\partial T}\right|$ such that $T^* > T_{\max}$.
4: Cluster the data by: $x_i \in \widehat{\Omega}$, $\forall x_i$ s.t. $\phi_n(x_i) > T^*$.
```

2.1 Outlier seed detection via Random sampling

In some applications the set of initial seeds Ω_o might prove to be errornous, as it may contain a subset of *false* samples Ω_o^f, such that $\Omega_o = \Omega_o^f \cup \Omega_o^t$, $\Omega_o^f \notin \widehat{\Omega}$ and $\Omega_o^t \in \widehat{\Omega}$. Algorithm 1 can not identify these seeds, as their diffusion values are set a-priory to 1.0. Thus, a sample can be analyzed only if it is not used as a seed. Hence, we propose to choose B random subsets of Ω_o, denoted $\left\{\Omega_0^b\right\}$, and propagate from them using Algorithm 1. Let the corresponding set of diffusions be $\left\{\phi_n^b\right\}_{b=1}^{B}$. We average $\left\{\phi_n^b\right\}$ over b, where for each sample we exclude the instances where it was used as a seed. This corresponds to analyzing the seeds using only secondary evidence. This is summarized in Algorithm 3.

2.2 K-LSA clustering

In this section we describe an unsupervised clustering scheme that partitions the input dataset into a given number of K clusters. This scheme is based on the LSA and denoted as K-LSA. The K-LSA partitions the data by applying the LSA classifier K times, using K sets of seeds $\left\{\Omega_o^1, \Omega_o^2, \ldots, \Omega_o^K\right\}$. Each of the sets$\left\{\Omega_o^k\right\}$ is propagated separately using Algorithm 1. Then we determine the classification of each sample by maximizing its diffusion value. This is summarized in Algorithm 4.

3 Applications and Experimental results

In this section we apply the proposed LSA and K-LSA schemes to the analysis of different data sources. In Section 3.1 we present image segmentation results using different cues and exemplify the LSA's ability to

Algorithm 3 Random sampling

```
1: Given the set of seeds $\Omega_o$, the random sampling ratio $r$ and the number of random iterations $B$.
2: for $b = 1$ to $B$ do
3:     Choose a random set $\Omega_0^b \subset \Omega_o$ made of $r \cdot |\Omega_o|$ elements and define the function $Q^b(x)$ such that
           $Q^b(x) = \begin{cases} 1 & x \in \Omega_0^b \\ 0 & otherwise \end{cases}$
4:     Apply Algorithm 1 using $\Omega_0^b$ as seeds and denote the result $\phi_n^b$.
5: end for
6: Return the weighted average
           $\overline{\phi_n}(x) = \frac{\sum\limits_{b=1..B} \phi_n^b(x) Q^b(x)}{\sum\limits_{b=1..B} Q^b(x)}.$
```

Algorithm 4 $K - LSA$ partitioning

```
1: Given $K$ sets of seeds $\left\{\Omega_o^1, \Omega_o^2, \ldots, \Omega_o^K\right\}$.
2: Apply Algorithm 1 separately for $\left\{\Omega_o^1, \Omega_o^2, \ldots, \Omega_o^K\right\}$, and denote $\phi_n^k$ as the result of propagating $\Omega_0^k$.
3: The corresponding cluster $C_k$ for a sample $x_i$ is given by
           $k = \arg\max_{\widetilde{k}} \left\{\phi_n^{\widetilde{k}}(x_i)\right\}.$
```

handle errornous seeds. Images are structured data sources, where each sample (pixel) is related to limited, a-priory known number of pixels. We consider non-structured data in Section 3.2, where we apply the LSA and K-LSA to documents classification. This is also an example of a directed graph that is handled by the LSA and K-LSA without making any adjustments. Last, we present a dynamic, example based search algorithm in Section 3.3, for the retrieval of images from databases.

3.1 Image segmentation

We start by presenting a series of segmentations of the image given in Fig. 3, where we used the YUV color space as a feature space and the Gaussian as a kernel. The distance metric is computed within a 7×7 image patch centered around each pixel. This results in a block-diagonal distance and affinity matrices. We also show that the maxima and corresponding saddle points of $\left|\frac{\partial N(T)}{\partial T}\right|$ are useful for partitioning, as suggested in Section 2 and Algorithm 1.

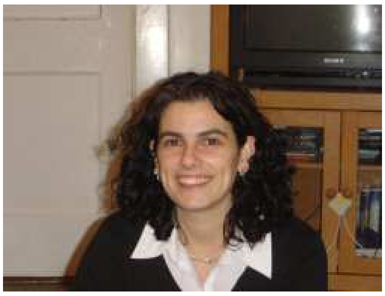
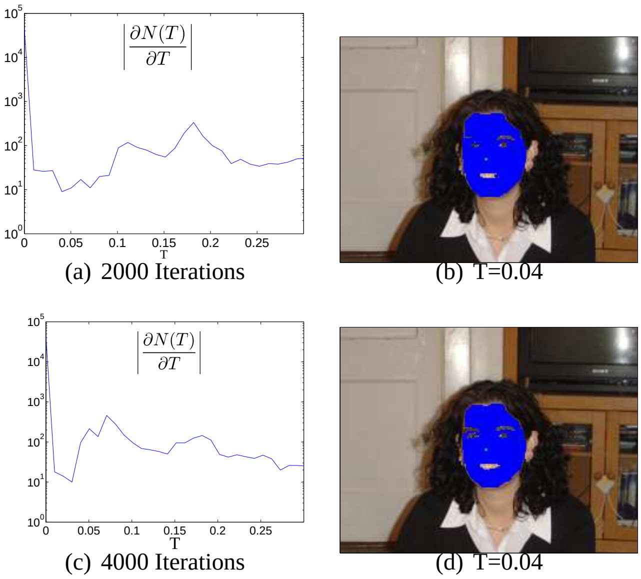


Figure 3: The image used for the segmentations using the YUV color space.

Figure 4 shows the results of applying the proposed scheme using a varying number of iterations and a scale factor of $\varepsilon = 5$. The threshold T^* is detected using Algorithm 2 (see Figs. 4a and 4c) and the segments are then identified by displaying the pixels for which $\phi_n > T^*$. Note that the saddle points of $\left|\frac{\partial N(T)}{\partial T}\right|$ correspond to the same image segments depicted in Figs. 4b and 4d.The outcome of changing the scale factor ε on the solution is studied in Fig. 5. We see that for both $\varepsilon = 10$ and $\varepsilon = 20$, we get similar clustering results. Note that due to increasing the scale factor from $\varepsilon = 5$ (as it was used in Fig. 4), we can no longer distinguish between fine details of the face.



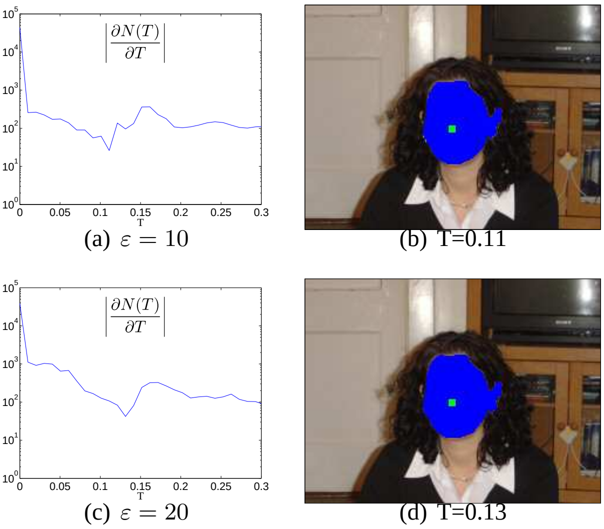
(a) 2000 Iterations

(b) T=0.04

(c) 4000 Iterations

(d) T=0.04

Figure 4: LSA segmentation results for different numbers of iterations. The results in (a) and (b) correspond to using 2000 steps, while those in (c) and (d) correspond to 4000. Note that for both cases we achieve similar segmentation results.



(a) $\varepsilon = 10$

(b) T=0.11

(c) $\varepsilon = 20$

(d) T=0.13

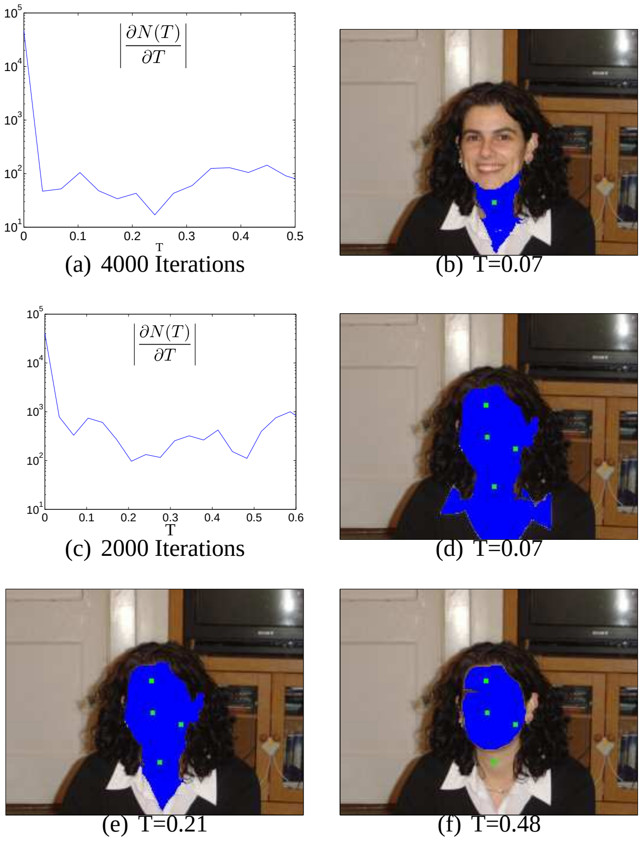
Figure 5: LSA segmentation results for varying values of the scale factor ε. (a) and (b) correspond to $\varepsilon = 10$, while (c) and (d) correspond to $\varepsilon = 20$. For both settings we achieve similar clustering results.

Next, we studied the sensitivity of the LSA to errornous seeds. This issue is essential in many real-world applications such as protein data analysis. We start by showing that the resulting classification is robust to a small number of errornous seeds. Then we apply the random sampling procedure described in Section 2.1 to identify the erroneous seeds. The experimental setup is depicted in Fig. 6, where we have four seeds, three of which are placed on the face (the first cluster), and the fourth one is in the neck area (the second cluster). Figures 6a-6b show that the neck area is indeed a different segment than the face area. We aim to show that using the LSA, we will still be able to cluster the face area.For that we applied random sampling (Algorithm 3). The results given in Table 1 show that the point on the neck has a significantly lower diffusion value.

Point	A	B	C	D
$\overline{\phi_n}\,(x)$	0.3992	0.5957	0.5352	0.1750

Table 1: Random sampling probabilities for the setup given in Fig. 6. The points correspond to the points in Fig. 6 sorted according to a descending order of height. A is the highest point in Fig. 6 (forehead) and D is the lowest (neck area).

We also applied the LSA to image segmentation using other cues. Figures 7a-7d show image segmentations using the RGB color space, while in Figs. 7e-7g we used the variance of 5×5 image patches around each pixel as a cue. In all cases we used $\varepsilon = 10$ and 500 iterations.



(a) 4000 Iterations

(b) T=0.07

(c) 2000 Iterations

(d) T=0.07

(e) T=0.21

(f) T=0.48

Figure 6: LSA clustering with errornous seeds. We aim to cluster the face as in Fig. 5. The image is clustered using 4 seeds. The outlier seed is located on the woman's neck and results in the segmentation shown in (b) and (c). Using the other 3 seeds (shown in (e)), we get the segmentations shown in (e). By lowering the threshold both clusters are unified as in (f).

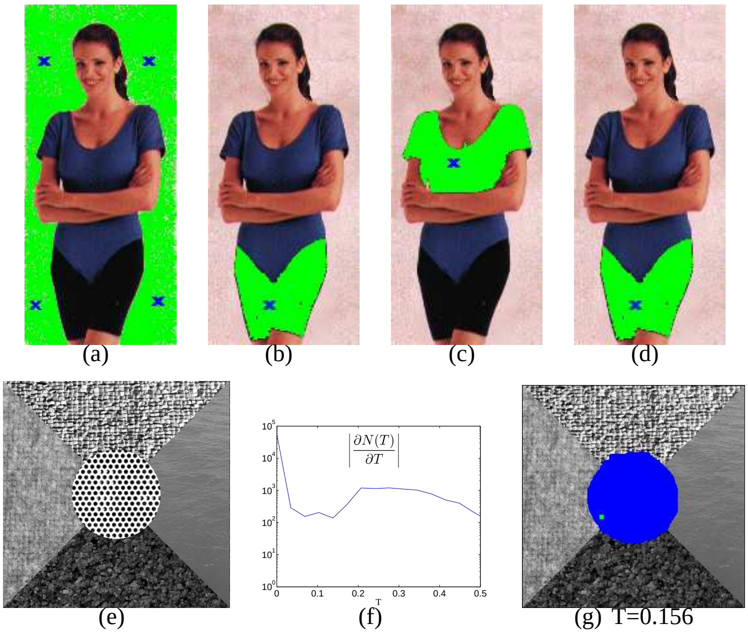


Figure 7: LSA based image segmentation using the RGB color space and patch variance as feature spaces. The seeds are marked by the X sign. Figures (a)-(d) show the RGB based clustering, and the clustered segments are depicted by the green overlay. Figures (e)-(g) show the clustering of a texture image, the corresponding $\left|\frac{\partial N(T)}{\partial T}\right|$ and the clustering results, respectively.

3.2 Textual data: example based search

Documents search is a common application in machine learning and in this section we direct our attention to example based article search. We applied the LSA to derive two document search schemes: in the first, the user specifies a query by pointing out a few sample documents in the category of interest, and we aim to identify the remaining documents in the category. The second, aims to partition the entire corpus of documents using the unsupervised K-LSA scheme presented in Section 2.2. This allows us to compare our approach to previous results that dealt with unsupervised partitioning.

We utilized standard text corpuses with known clusters. For each corpus we computed the Mutual information (MI) document-term matrix suggested in [26]. Let R be a document-term matrix, where $r\left(d,w\right)$ is the number of instances of a word w in the document d. The MI doc-term matrix is computed by:

1. Normalize the corpus size by computing $\widetilde{r}\left(d,w\right) = r\left(d,w\right) / \sum\limits_{d,w} r\left(d,w\right).$
2. Compute $E_w\left(d\right) = \sum\limits_{w} \widetilde{r}\left(d,w\right)$ and $E_d\left(w\right) = \sum\limits_{d} \widetilde{r}\left(d,w\right).$
3. The Mutual information document-term matrix is given by $\widehat{r}\left(d,w\right) = \log\left(\frac{\widetilde{r}(d,w)}{E_w(d)E_d(w)}\right).$

Given $\widehat{r}\left(d,w\right)$, we retained the 7 nearest neighbors (NN) for each document. Although this results in an asymmetric doc-term matrix, both the LSA and K-LSA were applied as is. In general, applying a NN search allows to handle situations, where the samples within a cluster are not well differentiated from the ones outside of it. The analysis in Section 2 is irrelevant here, as it is based on a symmetric energy function U, and we reserve the analysis of the asymmetric case for future work (Please note the discussion in Section 4).

For each corpus we use the LSA to cluster each of its a-priori known document categories, using a varying number of seeds. For a given classification, its accuracy is measured in terms of the area of the ROC curve [27]. We choose a set of thresholds $\{T_i\}$, uniformly spread over the interval $[0..1]$, and for each threshold positively identify the samples for which $\phi_n > T_i$ as belonging to the cluster of interest. Using the reference categorization, we identify the true and false positives and compute the area under the ROC curve. It is well known [27] that a random classifier would yield an area of 0.5, while an optimal classifier yields an area of 1.0.

To asses the accuracy of the unsupervised K-LSA classification, and compare it to the comprehensive clustering results given in [28], we adopt the Normalized Mutual Information (NMI) measure [29], as a distance measure between the K-LSA and the reference partitionings:

$$NMI = \frac{\sum\limits_{h,l} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h n_l}\right)}{\sqrt{\sum\limits_{h} n_h \log\left(\frac{n_h}{n}\right) \sum\limits_{l} n_l \log\left(\frac{n_l}{n}\right)}}. \tag{3.1}$$

n_h is the number of documents in class h, n_l the number of documents in cluster l and $n_{h,l}$ is the number of documents in class h as well as in cluster l. The denominator term normalizes this measure to lie in the range [0,1], making it invariant to the size of the analyzed set. The NMI value is 1.0 when the clustering results perfectly match the external category labels and close to zero for a random classification.

We start with the *Science News* corpus. This is a heterogeneous corpus of text documents obtained from the Science News web site. This corpus consists of 1047 documents in eight classes. We run Algorithm 1 using different numbers of seeds, ranging from 1 to 10. For each number of seeds, we repeated the simulations 1000 times and report the averages and standard deviations of the ROC area. These results are shown in Table 2 and we see that the ROC areas are close to 1.0 for all article categories except for the *Life Sciences*.

Documents classes	No. docs	No. Seeds 1	3	5	7	9
Anthropology	54	0.81 ± 0.13	0.90 ± 0.04	0.92 ± 0.03	0.92 ± 0.01	0.93 ± 0.01
Astronomy	121	0.91 ± 0.16	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.99 ± 0.01
Behavioral Sciences	72	0.84 ± 0.14	0.92 ± 0.03	0.93 ± 0.01	0.94 ± 0.01	0.94 ± 0.01
Earth Sciences	137	0.73 ± 0.16	0.86 ± 0.06	0.88 ± 0.05	0.90 ± 0.03	0.91 ± 0.03
Life Sciences	205	0.64 ± 0.09	0.69 ± 0.08	0.73 ± 0.06	0.74 ± 0.05	0.76 ± 0.05
Math & CS	60	0.91 ± 0.10	0.95 ± 0.02	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.01
Medicine	280	0.85 ± 0.14	0.91 ± 0.04	0.92 ± 0.02	0.92 ± 0.01	0.92 ± 0.01
Physics	118	0.81 ± 0.13	0.85 ± 0.08	0.88 ± 0.05	0.90 ± 0.03	0.90 ± 0.03

Table 2: LSA Classification results for the *Science News* corpus. The results are given in terms of the averages and standard deviations of the area of the ROC curves (over 1000 simulations), as a function of the number of seeds used.

The ROC curve was also used to asses the clustering scheme in Algorithm 2, and the use of the saddle point of $\left|\frac{\partial N(T)}{\partial T}\right|$ as the clustering threshold. This is exemplified in Fig. 8, where we applied the clustering scheme to the *Anthropology* (54 articles) and *Astronomy* (121 articles) categories within the *Science News* corpus. For each of them we present the results of a single diffusion run using 10 randomly chosen seeds. Figures 8a and 8c depict $\left|\frac{\partial N(T)}{\partial T}\right|$ for the *Anthropology* and *Astronomy* categories, respectively. We identify T^* as the first saddle point following the maximum of $\left|\frac{\partial N(T)}{\partial T}\right|$, and present the corresponding ROC curves in Figs. 8b and 8d, where each point on these curves corresponds to a particular choice of a threshold T^*. The 'sweet spot' on a ROC curve is located as close as possible to its upper-left corner, where ideally, one achieves 100% true positives and 0% false positives. Turning our attention to Figs. 8b and 8d we see that in both cases, the chosen threshold T^* is closed to the optimal choice.

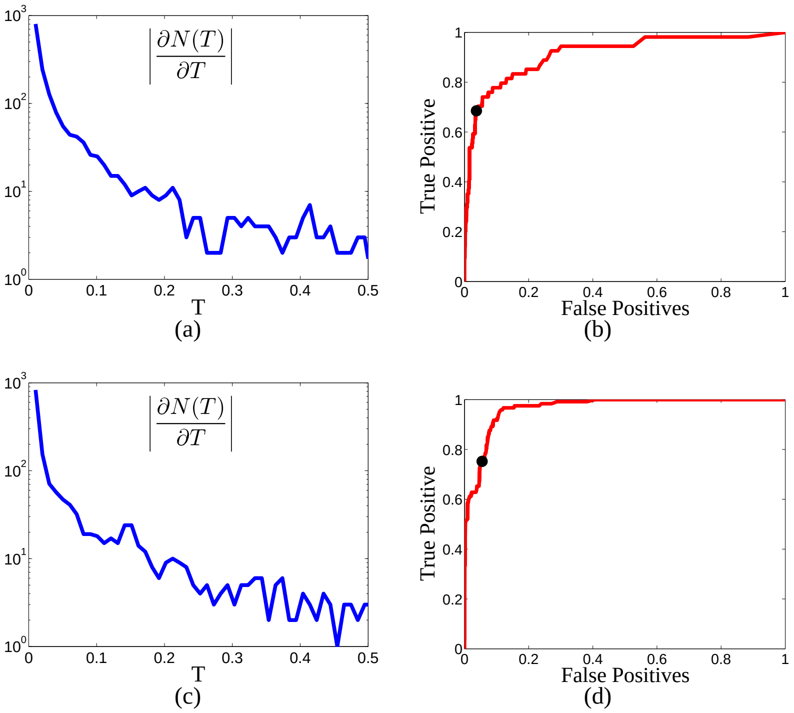


Figure 8: The classification accuracy of the Local spectral analysis scheme. (a) and (b) depict the function $\left|\frac{\partial N(T)}{\partial T}\right|$ and the ROC curve corresponding to a single run of the LSA, using 10 seeds extracted from the *Anthropology* category of the *Science news* corpus. The saddle point is detected at $T^* = 0.15$ and the corresponding True Positives and False Positives are overlayed on the ROC curve in (b). This choice of T^* is close to the optimal choice in the upper-left corner of the ROC curve. Similar results are shown for the *Astronomy* category in (c) and (d), for which $T^* = 0.13$.

In order to compare the K-LSA results to former schemes, we used the set of corpuses provided in

Zhong's work [28] and available as doc-term matrices in Matlab format[1]. In [28], Zhong provides the unsupervised classification results of 14 state-of-the-art schemes that we used for comparison.

We consider the NG17-19 dataset that is a subset of the NG20 corpus of newsgroups messages. The NG17-19 contains messages from each of the three categories on different aspects of politics. These categories are expected to be difficult to separate and the set consists of 2,998 documents in a 15,810 dimensional vector space. The K-LSA results are shown in Fig. 9, and are comparable to the best scheme in [28] (the CLUTO algorithm). Note that, at most, we have only used 30 (1%) of the 2998 articles as seeds. For the LSA, we report the ROC areas as a function of the number of seeds in Table 3. For these simulations, we randomly sampled each set of seeds a 1000 times and present the average and standard deviation of the ROC areas.

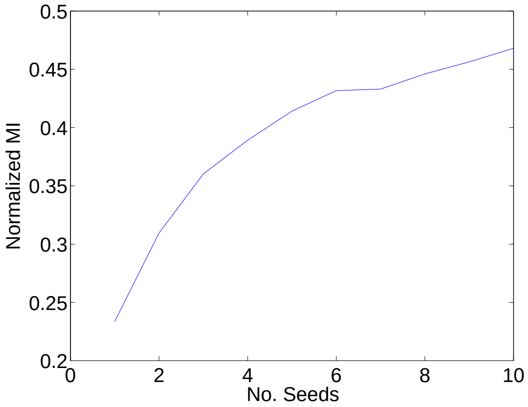


Figure 9: K-LSA unsupervised clustering results for the NG17-19 corpus (2998 documents). We show the Normalized Mutual Information (NMI) between the K-LSA and reference partitionings. The K-LSA outperforms the best result in [28] (0.46) using 1% of the documents as seeds.

Finally, in order to further asses the performance of the K-LSA, we utilized the $tr11$, $tr23$, $tr41$ and $tr45$ datasets. These are derived from the TREC collections (http://trec.nist.gov) and we followed the same experimental setup as before. For each set we compute the NMI for varying numbers of seeds and compare it to the best classification given in [28]. We present these results in Table 4. In all of the articles categories, the K-LSA achieved comparable accuracy to the best unsupervised scheme, using $5\% - 10\%$ of the samples as seeds.

[1] http://www.cse.fau.edu/~zhong/software/index.htm.

C	$\|C\|$	Number of seeds 1	3	5	7	9
“17”	1000	0.64 ± 0.13	0.84 ± 0.09	0.88 ± 0.06	0.90 ± 0.03	0.91 ± 0.03
“18”	999	0.62 ± 0.11	0.78 ± 0.11	0.87 ± 0.07	0.88 ± 0.06	0.91 ± 0.04
“19”	999	0.60 ± 0.10	0.71 ± 0.10	0.77 ± 0.08	0.79 ± 0.07	0.81 ± 0.07

Table 3: LSA classification results for the NG17-19 corpus. The LSA was used to separately cluster each of the document categories. We present the average ROC curve areas (over a 1000 simulations) and the corresponding standard deviations. $|C|$ is the number of documents in each category.

				Number of seeds				
C	$C\#$	$\overline{\|C\|}$	Best of [28]	1	3	5	7	9
$tr11$	9	46	0.68	0.35 ± 0.06	0.46 ± 0.05	0.52 ± 0.04	0.57 ± 0.03	0.61 ± 0.03
$tr23$	6	34	0.43	0.33 ± 0.06	0.45 ± 0.06	0.52 ± 0.06	0.58 ± 0.06	0.64 ± 0.06
$tr41$	10	88	0.67	0.51 ± 0.04	0.60 ± 0.04	0.64 ± 0.03	0.68 ± 0.03	0.70 ± 0.03
$tr45$	10	69	0.5	0.46 ± 0.05	0.53 ± 0.04	0.58 ± 0.03	0.62 ± 0.03	0.66 ± 0.03

Table 4: K-LSA Partitioning accuracy results, measured in terms of the Normalized Mutual Information (NMI), for the $tr11$, $tr23$, $tr41$ and $tr45$ article corpuses. $C\#$ is the number of categories in each corpus C, and $\overline{|C|}$ is the average number of documents in each category.

3.3 Dynamic Visual Search

Dynamic search refers to a scheme that searches for the members of a particular sub-cluster within a larger cluster, while agglomerating information over time. The work by Avraham and Lindenbaum [30, 31] presents and analyzes such a framework for image search and retrieval from databases. The external information source is denoted as the *Oracle* and it provides a binary answer to the query whether a particular sample x_i belongs to the cluster $\widehat{\Omega}$ or not. Given a database Ω of images, we aim to extract all the samples related to a particular cluster $\widehat{\Omega} \subset \Omega$, while minimizing the number of calls to the Oracle.

In practice, the Oracle may correspond to a user searching for a particular image within a database and providing yes/no instructions to the image retrieval software. Such an approach overcomes the need to define a formal language for image retrieval and allows the user to query by the equivalent of the intuitive statement: 'give me an image similar to this one'.

Avraham and Lindenbaum propose a linear estimation search algorithm, denoted VSLE (Visual Search by Linear Estimation), which utilizes cross-image similarity measures and propagates the binary vector of the probabilities of a certain sample x_i to belong to $\widehat{\Omega}$. The search terminates when all of the elements in $\widehat{\Omega}$ are verified by the Oracle.

We applied the LSA scheme to the dynamic search task, and similarly to [30], conclude that the search process should be divided into two parts:

1. **Searching for the first element in $\widehat{\Omega}$:** This is a data rejection task, where given a sample $x_i \in \widetilde{C}$ and $\widetilde{C} \neq \widehat{\Omega}$, we aim to identify all the other samples in $\widetilde{C}$ using the LSA, without having to use the Oracle for that. Then we choose the next sample to be verified by the Oracle, to be as far as possible from the set $\widetilde{C}$. By that, we utilize the manifold structure of the clusters $\widetilde{C}_i$ where $\Omega = \bigcup\limits_i \widetilde{C}_i + \widehat{\Omega}$. This scheme is summarized in Algorithm 5.

2. **Identifying $\widehat{\Omega}$ given a sample in it:** Given a single element $x_i \in \widehat{\Omega}$, we aim to propagate the similarity to all of the other elements within $\widehat{\Omega}$ and verify each of them using the Oracle. Hence, in each step we propagate the information from the set of verified samples $\{x_i\} \in \widehat{\Omega}$ to the rest of the elements in $\widehat{\Omega}$. The next sample is then given by x_i such that $i = \arg\max\limits_{\widetilde{i}} \phi_n \left(x_{\widetilde{i}}\right)$. This is summarized in Algorithm 6.

The LSA based retrieval scheme was tested by averaging 1000 repetitions of Algorithms 5 and 6. For that we utilized the same distance matrices as the ones used in [30] (Courtesy of Tamar Avraham and Michael

Algorithm 5 Finding the first element in $\widehat{\Omega}$

```
1: Given the set $\Omega$ compute the corresponding random walk matrix $M$.
2: Randomly choose a sample $x_0 \in \Omega$ and initiate the set $E = x_0$.
3: while $E \cap \widehat{\Omega} = \varnothing$ do
4:     Apply Algorithm 1 using $E$ as the set of seeds.
5:     Choose the sample $x_i$ such that $i = \arg\min_{\tilde{i}} \phi_n\left(x_{\tilde{i}}\right)$. $x_i$ is the sample farthest away from the set $E$.
6:     Ask the Oracle if $x_i \in \widehat{\Omega}$, if so go to Step#8, otherwise add $x_i$ to $E$.
7: end while
8: Return $x_i \in \widehat{\Omega}$.
```

Algorithm 6 Identifying $\widehat{\Omega}$ given its first element

```
1: Given the set $\Omega$ compute the corresponding random walk matrix $M$.
2: Form the set $E = x_0$ where $x_0 \in \widehat{\Omega}$.
3: while $E \neq \widehat{\Omega}$ do
4:     Apply Algorithm 1 using $\widetilde{C}$ as the set of seeds.
5:     Choose the sample $x_i$ such that $i = \arg\max_{\tilde{i}} \phi_n\left(x_{\tilde{i}}\right)$. $x_i$ is the sample closest to the set $E$.
6:     Ask the Oracle if $x_i \in \widehat{\Omega}$, if so add it to $E$.
7: end while
8: Return the set $\widehat{\Omega} = E$.
```

Lindenbaum). Table 5 summarizes the experimental results for four different visual databases. Compared to the textual search described in Section 3.2, the sets are significantly smaller and are of $O\left(10\right)$. The best results are achieved for the *cars* and *faces* databases where the total improvement is 33% and 50%, respectively. For the *elephants* and *parasols* databases the VSLE prevailed. We attribute that to the small size of these databases, 24 and 40 elements, respectively, and conclude that the LSA is better suited for large datasets, where the manifold structure comes into play.

4 Conclusions and Future Work

In this work we presented a computational approach to local spectral data analysis. Our approach propagates local information, based on a given set of samples, denoted as seeds. We showed that by choosing a set of seeds from a particular cluster, we are able to derive a clustering scheme, that detects the discontinuities in

Database			Initial choice			Search		Overall	
Ω	$\|\Omega\|$	$\|C\|$	Random	VSLE	LSA	VSLE	LSA	VSLE	LSA
cars	100	10	10	6.69	3.74 ± 4.03	26.18	17.44 ± 1.09	32.87	21.18 ± 3.93
elephants	24	4	6	2.95	2.74$\pm$1.72	3.59	4.48 ± 1.00	6.54	7.63$\pm$1.02
faces	146	7	20.85	17.43	14.25$\pm$6.41	45.57	20.44 ± 1.69	63	34.63 ± 5.1
parasols	30	6	5	2.33	2.94 ± 1.02	5	5.50 ± 0.50	7.33	7.9 ± 1.02

Table 5: Dynamic visual retrieval results. For each database Ω, we present the average number and standard deviation (over a 1000 simulations) of Oracle calls, needed to positively identify all of the items in it. $|\Omega|$ is the number of objects in each database, and $|C|$ is the number of the categories of objects in each database Ω. The *random* column refers to the number of oracle calls needed when the samples are randomly picked.

the diffusion function. We effectively applied the proposed scheme to data sources such as images, document corpuses and visual databases. We analyzed our approach based on the Fokker-Planck interpretation of spectral embedding.

In future, we intend to extend our work on Content based image retrieval (CBIR). This field has recently gained much attention and we aim to process larger image datasets. In addition, following [30], we allowed no classification errors, and thus each target sample had to be positively identified by the Oracle. This led to the nearest neighbor sampling strategy in Step #5 of Algorithm 6. A different strategy might lead to classification errors, but might also significantly reduce the number of Oracle calls.

We also intend to analyze directed/asymmetric graphs/networks. Although we presented promising experimental results for the document mining task using such graphs, and these appear naturally in application fields such as communication and biological networks, the Fokker-Planck based analysis is inapplicable to them. Our approach would be to introduce a pressure term into the Fokker-Planck equation in Eq. 2.1 to break down its symmetry.

As an image segmentation scheme, the LSA is purely data driven and does not incorporate image domain geometric properties such as curvature. Those were found beneficial in prior region growing schemes such as [32] and [33]. Incorporating such information would require temporal analysis of the evolution of the diffusion ϕ_n. A related task is to derive a narrow-band like formulation of the LSA. The narrow-band is an efficient approach to computing approximate solutions of the heat equation [32], where at each time step, the solution is computed within a narrow region around the wave front of the heat equation's solution. If applicable for the LSA, such an approach might enable us to better analyze large and massive datasets.

5 Acknowledgments

The author would like to thank Amit Singer (Applied Math department, Yale University) for many helpful discussions during the course of this work, regrading the Fokker-Planck operator and pointing out the works by Schuss and Matkowsky. We also thank Tamar Avraham and Michael Lindenbaum (Computer Science Department, Technion IIT) for kindly providing their dynamic search data and motivating our work on dynamic visual search.

References

[1] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, 2000.

[2] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 6, no. 15, pp. 1373–1396, June 2003.

[3] D. Donoho and C. Grimes, “Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[4] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimension reduction via local tangent space alignement,” Department of computer science and engineering, Pennsylvania State University, Tech. Rep. CSE-02-019, 2002.

[5] R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[6] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[7] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7432–7437, May 2005.

[8] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators," in *Advances in Neural Information Processing*, 2005.

[9] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of markov chains," *The Annals of Applied Probability*, vol. 1, no. 1, pp. 36–61, 1991.

[10] F. Chung, *Spectral graph theory*. CBMS-AMS, May 1997, no. 92.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[12] Y. Weiss, "Segmentation using eigenvectors: A unifying view." in *ICCV*, 1999, pp. 975–982.

[13] M. Meila and J. Shi, "A random walk's view of spectral segmentation," *AI and Statistics (AISTATS)*, 2001.

[14] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning and data set parameterization," *IEEE Pattern Analysis and Machine Intelligence*, 2006.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 888–905, August 2000.

[16] M. Meila and J. Shi, "Learning segmentation by random walks." in *Advances in Neural Information Processing*, 2000, pp. 873–879.

[17] S. X. Yu and J. Shi, "Segmentation given partial grouping constraints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 173–183, 2004.

[18] R. Nock and F. Nielsen, "Grouping with bias revisited," vol. 2, June 2004, pp. 460–465.

[19] N. Tishby and N. Slonim, "Data clustering by markovian relaxation and the information bottleneck method." in *NIPS*, 2000, pp. 640–646.

[20] M. Szummer and T. Jaakkola, “Partially labeled classification with markov random walks,” in *Advances in Neural Information Processing*, T. D. et al., Ed., vol. 14. MIT Press, 2001.

[21] D. Harel and Y. Koren, “On clustering using random walks,” in *FST TCS ’01: Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science*. London, UK: Springer-Verlag, 2001, pp. 18–41.

[22] ——, “Clustering spatial data using random walks,” in *KDD ’01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2001, pp. 281–286.

[23] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, “Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[24] Z. Schuss and B. J. Matkowsky, “The exit problem: a new approach to diffusion across potential barriers,” *SIAM Journal on Applied Mathematics*, vol. 35, no. 3, pp. 604–623, June 1979.

[25] B. J. Matkowsky and Z. Schuss, “Eigenvalues of the fokker-planck operator and the approach to equilibrium for diffusions in potential fields,” *SIAM Journal on Applied Mathematics*, vol. 40, no. 2, pp. 242–254, April 1981.

[26] C. E. Priebe, D. J. Marchette, Y. Park, E. ward J. Wegman, J. L. Solka, D. A. Socolinsky, D. Karakos, K. W. Church, R. Guglielmi, R. R. Coifman, D. Lin, D. M. Healy, M. Q. Jacobs, and A. Tsao, “Iterative denoising for cross-corpus discovery,” in *COMPSTAT: Proceedings in Computational Statistics*. Physica-Verlag, Springer., 2004, p. 381–392.

[27] T. Fawcett, “Roc graphs: Notes and practical considerations for data mining researchers,” *Technical Report HPL-2003-4, HP Labs, 2003.*

[28] S. Zhong and J. Ghosh, “Generative model-based document clustering: a comparative study,” *Knowledge and Information Systems*, vol. 8, no. 3, pp. 374–384, 2005.

[29] A. Strehl and J. Ghosh, “Cluster ensembles — a knowledge reuse framework for combining multiple partitions,” *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.

[30] T. Avraham and M. Lindenbaum, “Attention-based dynamic visual search using inner-scene similarity: Algorithms and bounds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 251–264, 2006.

[31] ——, “Dynamic visual search using inner-scene similarity: Algorithms and inherent limitations.” in *European Conference on Computer Vision*, vol. 2, 2004, pp. 58–70.

[32] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, “Fast geodesic active contours.” *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1467–1475, 2001.

[33] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.