Factorized Diffusion Map Approximation

Saeed Amizadeh
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15213

Hamed Valizadegan
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15213

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15213

Abstract

Diffusion maps are among the most powerful Machine Learning tools to analyze and work with complex high-dimensional datasets. Unfortunately, the estimation of these maps from a finite sample is known to suffer from the curse of dimensionality. Motivated by other machine learning models for which the existence of structure in the underlying distribution of data can reduce the complexity of estimation, we study and show how the factorization of the underlying distribution into independent subspaces can help us to estimate diffusion maps more accurately. Building upon this result, we propose and develop an algorithm that can automatically factorize a high dimensional data space in order to minimize the error of estimation of its diffusion map, even in the case when the underlying distribution is not decomposable. Experiments on both the synthetic and real-world datasets demonstrate improved estimation performance of our method over the standard diffusion-map framework.

1 Introduction

The emergence of complex high-dimensional datasets in recent years has spurred the interest of the machine learning community in manifold analysis and spectral algorithms. Laplacian-based methods, especially diffusion maps, are amongst the most popular approaches to analyze such data. A diffusion map defines a lower dimensional embedding of the data that preserves the cluster structure of the data at different resolutions. This methodology has been successfully applied to a variety of clustering and semi-supervised learning tasks [1,17,18,23–27].

Recent theoretical studies have shown that under some reasonable regulatory conditions, the data-based diffusion (Laplacian) matrix asymptotically converge to certain operators defined based on the true distribution of data [10,13,19,24]. In fact, the eigenfunctions of these asymptotic operators encode the cluster structure of the underlying density of data. An important question is how well we can estimate these eigenfunctions from a finite sample. Unfortunately it has turned out that the rate of convergence of the finite-sample approximations to true eigenfunctions is exponential in the dimension of the data, hence the problem suffers from the curse of dimensionality [13].

One possible way to alleviate the curse of dimensionality problem is based on the factorization of the input space into independent subspaces. Nadler et al. [15] showed that when the underlying distribution is decomposable, it leads to the decomposition of the diffusion eigenfunctions. This idea was later used by Fergus et al. [8] to implement scalable semi-supervised learning in large-scale image datasets. However, the scope of that work is somewhat limited. First, it assumes fully decomposable distributions. Second, it does not provide any insight on the quality of eigenfunctions built using the decomposition.

The objective of this paper is to study how the factorization of the underlying distribution into independent subspaces can help us to approximate the true eigenfunctions from a finite sample more accurately. We show that if the underlying distribution is factorizable, we can get significant reductions in the error bound for estimating the major eigenfunctions. In fact, this is analogous to machine learning criteria and models that rely on the underlying distribution structure to compensate for the insufficient sample size.

To clarify this idea, consider the synthetic 3D dataset shown in Figure 1 with four clusters. The first row

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22. Copyright 2012 by the authors.

in the figure shows the four major diffusion eigenfunctions of a sample with 1000 points where the color code shows the sign of the eigenfunctions. Each eigenfunction effectively separates the clusters from a different angle so that as a whole these eigenfunctions discovers the overall cluster structure in the data. Now, if we decrease the sample size to a half, the eigenfunctions and therefore the clustering result will be perturbed (as shown in the second row). However, this specific dataset is generated in a way that the Z coordinate is independent of X and Y. If we incorporate this information using the framework proposed in this paper, we get the same result as the first row but now only with 500 points (as shown in the third row).

As the above example shows, having a factorizable underlying distribution can speed up the convergence of the empirical eigenvectors to true eigenfunctions. However, the distributions of real-world data may not factorize into independent subspaces. In this case, the key question is how much error on diffusion eigenfunctions is introduced if we *impose* such independence assumptions upon non-decomposable distributions. This idea is similar to imposing structural assumptions for model selection in order to decrease the parameter learning complexity. In this paper, we study the tradeoff between the estimation error of diffusion eigenfunctions and the approximation error introduced by imposing the independence assumptions on the underlying distribution. We propose and develop a greedy algorithm which considers different independence assumptions on the distribution in order to find a factorizable approximation that minimizes the error in estimates of diffusion eigenfunctions. To show the merits of our framework, we test it on clustering tasks using both synthetic and real-world datasets.

2 Related Work

Our work is related to a large body of existing work that utilize the Laplacian-based spectral analysis [5] of the similarity matrix of the data to find a low dimensional embedding of data. Such methods cover a wide range of learning tasks such as dimensionality reduction [2, 12], data clustering [1, 11, 17, 18, 24, 25], and semi-supervised learning [23, 26, 27]. While the focus of early works was more on developing new algorithms based on the spectral analysis of similarity metric [1,11,17,18,23–27], more recent works study the theoretical aspects of such analysis [3,10,13,15,19,20], leading to the development of the new algorithms [4,8].

Laplacian-based methods can be categorized into two major groups: locality-preserving methods [2] and diffusion maps [15], both of them are based on a similarity graph of data. These two groups differ in how they use the similarity metric. Locality-preserving methods aim at finding a mapping of data points to real values by minimizing the local variations of the mapping around the points. These methods can be considered as special cases of kernel PCA [13]. Diffusion maps are based on the random walk on the similarity graph of data and can be considered as non-linear version of Multidimensional Scaling (MDS) [7] technique. Many of these methods are built upon a similarity graph on the datapoints in the input space which has to be constructed based on the distance metric in the input space. One popular method to build the similarity graph is to transform the Euclidean distances into the similarity weights using the Gaussian similarity kernel with some bandwidth ε.

Several authors study the convergence rate of the laplacian-based methods, either by assuming that data lie exactly on a lower dimensional Riemannian manifold in the original space [3, 10, 19], or considering a general underlying distribution that generates data [8,13,24]. Our framework is built upon the result of the later group. For a fixed kernel bandwidth ε, von Luxburg et. al. [24] showed that the normalized Laplacian operator converges with rate $O(1/\sqrt{n})$. Lee and Wasserman [13] study the rate of convergence for $\varepsilon \to 0$ and large sample size n and showed that the optimal rate is dependent on d, the dimension of data. This result, i.e. the dependency of the convergence rate to d, is the main inspiration for our work.

3 Diffusion Framework

In this section, we overview the basic concepts of the diffusion map and its estimation complexity; interested readers may refer to [12,13,15] for further details.

Let the dataset $\mathcal{D} = \{x^{(1)}, \ldots, x^{(n)}\}$ be the instances of the random variables $V = \{X_1, \ldots, X_d\}$ sampled iid from the distribution P with compact support $\mathcal{X} \subset \mathbb{R}^d$ with a bounded, non-zero density p. Define the similarity kernel $k_\varepsilon(x, y) = \exp(-\|x - y\|_2^2/4\varepsilon)$ on $\mathcal{X}$. The *discrete-time diffusion operator* $A_{p,\varepsilon} : L^2(\mathcal{X}) \mapsto L^2(\mathcal{X})$ (where $L^2(\mathcal{X})$ is the class of functions f defined on $\mathcal{X}$ s.t. $\int f^2(x)dP(x) < \infty$) is defined as :

$$A_{p,\varepsilon}[f(x)] = \int_{\mathcal{X}} a_\varepsilon(x, y)f(y)p(y)dy, \qquad (1)$$

with asymmetric kernel $a_\varepsilon(x, y) = k_\varepsilon(x, y)/\int k_\varepsilon(x, z)p(z)dz$. $A_{p,\varepsilon}$ is a compact, positive operator with the largest eigenvalue $\lambda_{\varepsilon,1} = 1$ corresponding to the constant eigenfunction $\psi_{\varepsilon,1} = 1$. Alternatively, $A_{p,\varepsilon}$ can be represented using its eigen decomposition as $A_{p,\varepsilon} = \sum_{i=1}^{\infty} \lambda_{\varepsilon,i}\Pi_i$ where Π_i is the orthogonal projection on $\psi_{\varepsilon,i}$. Moreover, $a_\varepsilon(x, y)$ can be written as $a_\varepsilon(x, y) = \sum_{i=1}^{\infty} \lambda_{\varepsilon,i}\psi_{\varepsilon,i}(x)\varphi_{\varepsilon,i}(y)$ where

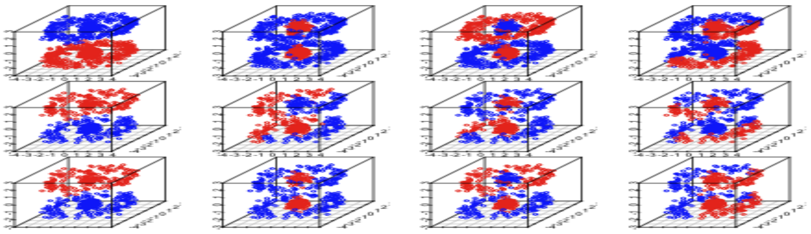

Figure 1: The synthetic 3D dataset with coordinate Z independent of X and Y

$\varphi_{\varepsilon,i}$ is the eigenfunction of $A^*_{p,\varepsilon}$, the adjoint of $A_{p,\varepsilon}$.

It is known that the principal eigenfunctions of $A_{p,\varepsilon}$ with the largest eigenvalues encode the manifold structure of data and therefore can be used for low dimensional embedding of the original data [12]. In fact, this is the basic motivation for introducing *diffusion map* $\phi_\varepsilon : \mathbb{R}^d \mapsto \mathbb{R}^r$ for $r < d$:

$$x \mapsto \phi_\varepsilon(x) = [\lambda_{\varepsilon,1}\psi_{\varepsilon,1}(x), \dots, \lambda_{\varepsilon,r}\psi_{\varepsilon,r}(x)]^T \qquad (2)$$

The underlying structure of data can be studied at different scales (like in hierarchical clustering). This gives rise to the notion of *m-step discrete-time diffusion operator* defined by exponenting the eigenvalues of $A_{p,\varepsilon}$ to the power of m as $A^m_{p,\varepsilon} = \sum_{i=1}^{\infty}\left(\lambda_{\varepsilon,i}\right)^m \Pi_i$. Subsequently, the asymmetric kernel for $A^m_{p,\varepsilon}$ is derived as $a_{\varepsilon,m}(x,y) = \sum_{i=1}^{\infty}\left(\lambda_{\varepsilon,i}\right)^m \psi_{\varepsilon,i}(x)\varphi_{\varepsilon,i}(y)$. $A^m_{p,\varepsilon}$ also induces the diffusion map $\phi^m_\varepsilon(\cdot)$ which maps the original data to a coarser cluster structure as m increases. Furthermore, one can extend the discrete scale of $A^m_{p,\varepsilon}$ (i.e. m steps of length ε) to the continuous scale $t = m\varepsilon$ (with $\varepsilon \to 0$) by defining the *continuous-time diffusion operator* $A^t_p : L^2(\mathcal{X}) \mapsto L^2(\mathcal{X})$ [13]:

$$A^t_p[f(x)] = \lim_{\varepsilon\to 0} A^{t/\varepsilon}_{p,\varepsilon}[f(x)] = \int_{\mathcal{X}} a_t(x,y)f(y)p(y)dy, \qquad (3)$$

where, $a_t(x,y) = \lim_{\varepsilon\to 0} a_{\varepsilon,t/\varepsilon}(x,y)$. The eigenvalues and the eigenfunctions of A^t_p are computed as: $\lambda^{(t)}_{p,i} = \lim_{\varepsilon\to 0}\lambda^{t/\varepsilon}_{\varepsilon,i}$ and $\psi^t_{p,i} = \lim_{\varepsilon\to 0}\psi_{\varepsilon,i}$, respectively.

The diffusion map induced by the eigen decomposition of A^t_p is a powerful tool for data embedding at different continuous scales t. However, in practice, we have to estimate the eigen decomposition of A^t_p from the finite sample $\mathcal{D}$ by computing the matrix $[\hat{A}_\varepsilon]_{n\times n} = [k_\varepsilon(x^{(i)}, x^{(j)})/\sum_{l=1}^{n} k_\varepsilon(x^{(i)}, x^{(l)})]_{n\times n}$ with the eigen decomposition $\hat{A}_\varepsilon \hat{u}_{\varepsilon,i} = \hat{\lambda}_{\varepsilon,i}\hat{u}_{\varepsilon,i}$. The empirical eigenfunctions are then computed from the eigenvectors $\hat{u}_{\varepsilon,i}$ using the Nyström approximation [13]:

$$\hat{\psi}_{\varepsilon,i}(x) = \frac{\sum_{j=1}^{n} k_\varepsilon(x, x^{(j)})\hat{u}_{\varepsilon,i}(x^{(j)})}{\hat{\lambda}_{\varepsilon,i}\sum_{j=1}^{n} k_\varepsilon(x, x^{(j)})} \qquad (4)$$

The eigenvalues and eigenfunctions of $\hat{A}_\varepsilon$ estimate their counterparts for $A_{p,\varepsilon}$ by estimating P with the empirical distribution $\hat{p} = 1/n$ (denoted by $\hat{A}_\varepsilon \to A_{p,\varepsilon}$ as $n \to \infty$) [13]. Since $\hat{A}^{t/\varepsilon}_\varepsilon \to A^{t/\varepsilon}_{p,\varepsilon}$ as $n \to \infty$, we can estimate the eigenspaces of A^t_p by those of $\hat{A}^{t/\varepsilon}_\varepsilon$. An important concern is how fast the rate of convergence is as $n \to \infty$. To answer this question, von Luxburg et. al. [24] showed that the normalized Laplacian operator converges with rate $O(1/\sqrt{n})$ given that ε *is fixed.* This result can be easily extended to the diffusion operator as well. The good thing about this rate is that it does not depend on the dimension d. However, to find the optimal trade-off between bias and variance, we need to let $\varepsilon \to 0$ as the sample size n increases. Lee and Wasserman [13] showed that the optimal rate for ε is $(\log n/n)^{2/(d+8)}$ and therefore the eigenfunctions converge as [9, 13]:

$$\|\psi^t_{p,i} - \hat{\psi}_{\varepsilon,i}\|_2^2 = O_P\left(\frac{t\sqrt{d}}{\mu^{(t)}_i}\left[\frac{\log n}{n}\right]^{2/(d+8)}\right) \qquad (5)$$

where, $\mu^{(t)}_i = \min_{2\le l\le i}\log(\lambda^{(t)}_{p,l-1}/\lambda^{(t)}_{p,l})$ is the multiplicative eigengap of A^t_p and $\|f\|_2^2 = \int_{\mathcal{X}} f^2(x)p(x)dx$. Unfortunately, this rate depends on the dimension exponentially which makes it a hard problem to estimate the eigenfunctions of A^t_p from a finite sample. Throughout the rest of this paper, we drop the subscript ε for the empirical operators assuming it is implicitly computed using the optimal rate above.

4 Factorized Diffusion Maps

4.1 The Factorized Approximation

Let $\boldsymbol{\mathcal{T}_k} = \{T_1, T_2, \dots, T_k\}$ be a partition of the variables in V into k disjoint subsets. Each T_i defines a subspace of V with dimension $d_i = |T_i|$. With a little abuse of notation, we also use T_i to refer to the subspace induced by the variables in T_i. We define the *marginal diffusion operator* $A^t_{p_i} : L^2(\mathcal{X}) \mapsto L^2(\mathcal{X})$:

$$\begin{gathered} A^t_{p_i}[g_z(x)] = \int_{T_i} a_t(x,y)g_z(y)p_i(y)dy, \\ x, y \in T_i, z \in V\backslash T_i \end{gathered} \qquad (6)$$

where $g_z(x) = f([x\ z]^T)$ assumes the variables in z are constants and p_i is the marginal distribution over the subspace defined by T_i. In other words, $A^t_{p_i}$ treats the input variables of $f(\cdot)$ which do not belong to T_i as constants. Furthermore, the partition $\boldsymbol{\mathcal{T}_k}$ defines the *factorized* distribution $q_{\boldsymbol{\mathcal{T}_k}} = \prod_{i=1}^k p_i$. To simplify the notation, for a fixed $\boldsymbol{\mathcal{T}_k}$, we refer to $q_{\boldsymbol{\mathcal{T}_k}}$ simply by q. We also define the *factorized diffusion operator* A^t_q the same as Eq. (3) with the true distribution p is replaced by the factorized distribution q. We have:

Lemma 1. *Let $\boldsymbol{\Lambda^t_i} = \{\lambda^{(t)}_{i,m} \mid 1 \leq m \leq \infty\}$ and $\boldsymbol{\Psi^t_i} = \{\psi^t_{i,m} \mid 1 \leq m \leq \infty\}$ be the set of eigenvalues and eigenfunctions of $A^t_{p_i}$, respectively. Then the sets:*

$$\boldsymbol{\Lambda^t_q} = \left\{ \prod_{i=1}^{k} \xi_i \mid \xi_i \in \boldsymbol{\Lambda^t_i} \right\}, \boldsymbol{\Psi^t_q} = \left\{ \prod_{i=1}^{k} \varphi_i \mid \varphi_i \in \boldsymbol{\Psi^t_i} \right\}$$

are respectively the eigenvalues and eigenfunctions of the factorized diffusion operator A^t_q.

Lemma 1 explicitly relates the eigenvalues and the eigenfunctions of the factorized diffusion operator A^t_q to eigenvalues and the eigenfunctions of the marginal diffusion operators $A^t_{p_i}$. We refer to the eigenvalues and the eigenfunctions based on the factorization as *multiplicative* eigenvalues and eigenfunctions.

The above decomposition also gives a recipe for computing the eigenfunctions of A^t_q from eigenfunction estimates in each subspace T_i. In particular, we can estimate the eigenfunctions in each subspace T_i independently and then multiply the results over all subspaces. This construction procedure is of special practical significance if p, in fact, factorizes according to $\boldsymbol{\mathcal{T}_k}$; that is, $p = q$. In that case, the principal eigenfunctions of A^t_p (with largest eigenvalues) can be estimated from a finite sample $\mathcal{D}$ (more accurately), if we make use of the fact that p is factorizable.

The multiplicative eigenvalue and eigenfunction estimates on the full variable space using q come with the following properties. First, the largest eigenvalue $\hat{\lambda}^{(t)}_{i,1}$ in each subspace T_i is $\hat{\lambda}^{(t)}_{i,1} = 1$ and is associated with the constant eigenfunction $\hat{\psi}^t_{i,1} = 1$. Therefore, the largest multiplicative eigenvalue of A^t_q (according to Lemma 1) will be $\hat{\lambda}_{q,1}(t) = \prod_{i=1}^k 1 = 1$ with a constant eigenfunction $\hat{\psi}^t_{q,1} = \prod_{i=1}^k 1 = 1$. Next, the second largest multiplicative eigenvalue of A^t_q will be $\hat{\lambda}^{(t)}_{q,2} = \hat{\lambda}_{j,2} \times \prod_{i \neq j} 1$ with the eigenfunction $\hat{\psi}^t_{q,2} = \hat{\psi}^t_{j,2} \times \prod_{i \neq j} 1$ where $j = \arg\max_r \hat{\lambda}^{(t)}_{r,2}$. That is, the second eigenfunction of A^t_q can be obtained from only one subspace (i.e. T_j) with a reduced dimensionality (d_j). Finally, the m-th multiplicative eigenvalue and eigenfunction $\hat{\lambda}_m(t)$ and $\hat{\psi}^t_m$ will be estimated using at most $\lg m$ marginal eigenfunctions on subspaces.

Lemma 2. *Suppose p factorizes according to $\boldsymbol{\mathcal{T}_k}$ and the eigen decomposition of A^t_p is constructed using the procedure suggested by Lemma 1 then the m-th eigenfunction of A^t_p associated with its m-th largest eigenvalue is the multiplication of the marginal eigenfunctions from "at most" $\min(k, \lceil \lg m \rceil)$ subspaces in $\boldsymbol{\mathcal{T}_k}$.*

From the estimation point of view, this result has a significant implication in that the estimation error of the m-th eigenfunction over the whole space can be reduced to the estimation error from at most $\lg m$ subspaces, each of which has a smaller dimension. To illustrate this, consider the second principal eigenfunction $\hat{\psi}^t_{q,2}$ described above. Since it depends only on one of the subspaces (with a reduced dimensionality), its rate of convergence, according to [13], should be faster. Hence its estimation error bound is reduced and equal to the error bound for that subspace. This observation further motivates the analysis of error for estimating the factorized diffusion map.

4.2 Error Analysis

In the previous subsection, we saw that if the underlying distribution p is factorizable, then we can decrease the estimation error bound of the m-th principal eigenfunction using the factorization to independent subspaces. However, in reality, p may not factorize at all; then, the question is how much error is introduced if we *approximate* p with q, and under which problem settings we get smaller error bounds by enforcing such a factorization. Suppose $\hat{A}^t_q$ is the estimated factorized diffusion operator from a sample of size n with the factorization according to $\boldsymbol{\mathcal{T}_k}$. Let $\psi^t_{p,m}$, $\psi^t_{q,m}$ and $\hat{\psi}^t_{q,m}$ represent the m-th eigenfunction of A^t_p, A^t_q and $\hat{A}^t_q$, respectively. We want to approximate $\psi^t_{p,m}$ with $\hat{\psi}^t_{q,m}$ and to study the error $\|\psi^t_{p,m} - \hat{\psi}^t_{q,m}\|^2_2$. We have the following inequality:

$$\mathcal{E}_{total}(q,m,t) \triangleq \|\psi^t_{p,m} - \hat{\psi}^t_{q,m}\|^2_2 \leq 2\|\psi^t_{p,m} - \psi^t_{q,m}\|^2_2 + 2\|\psi^t_{q,m} - \hat{\psi}^t_{q,m}\|^2_2 \qquad (7)$$

The first term on the right-hand side is the *approximation error or bias* which is due to approximating $\psi^t_{p,m}$ (i.e. p) with $\psi^t_{q,m}$ (i.e. q). Clearly, in case $p = q$, the approximation error is 0 and the inequality becomes an equality. Note that the approximation error is also a lower bound on $\mathcal{E}_{total}(q,m,t)$. The second term on the right-hand side is the *estimation error* of the factorized eigenfunction from the finite sample. We can bound these errors from above as follows:

Theorem 1. ***Upper bound on the approximation error:*** *Let $\sup_{f:\|f\|_2 \leq 1} \|f\|_\infty = \ell < \infty$, $\sup_{x,y} a_t(x,y) = \jmath < \infty$ and $\delta_m = \lambda^{(t)}_{p,m} - \lambda^{(t)}_{p,m+1}$ then*

$$\mathcal{E}_{app}(q,m,t) \triangleq \|\psi^t_{p,m} - \psi^t_{q,m}\|^2_2 \leq C \cdot D_{KL}(p\|q) \triangleq U_{app}(q,m,t) \qquad (8)$$

where $C = 32j^2\ell^2 \ln 2/\delta_m^2$ *and* $D_{KL}(\cdot\|\cdot)$ *denotes the Kullback-Leibler divergence.*

Theorem 1 translates the distance between the true and the approximated eigenfunction to the distance between the true underlying distribution and its factorized approximation.

Theorem 2. ***Upper bound on the estimation error:*** *Let* q *factorizes according to* $\boldsymbol{\mathcal{T}_k}$ *and* $\sup_{f:\|f\|_2\leq 1}\|f\|_\infty = \ell < \infty$. *Define* $S_m = \big\{(j_1,\ldots,j_k) \mid \forall i \in [1..k] : 1 \leq j_i \leq m$ *and* $\prod_{i=1}^k j_i \leq m\big\}$ *and the multiplicative eigengap* $\mu_{i,j}^{(t)} = \min_{2\leq l\leq j}\log(\lambda_{i,l-1}^{(t)}/\lambda_{i,l}^{(t)})$ *then we have:*

$$\begin{aligned}
&\mathcal{E}_{est}(q,m,t) \triangleq \|\psi^t_{q,m} - \hat{\psi}^t_{q,m}\|_2^2 \\
&\leq \max_{(j_1,\ldots,j_k)\in S_m} \ell^{2(k-1)} \sum_{i=1}^{k} 2^i \|\psi^t_{i,j_i} - \hat{\psi}^t_{i,j_i}\|_2^2 \\
&= O_P\Bigg(\max_{(j_1,\ldots,j_k)\in S_m} \ell^{2(k-1)} \sum_{\substack{i=1\\ j_i\neq 1}}^{k} \frac{2^i t\sqrt{d_i}}{\mu^{(t)}_{i,j_i}}\left[\frac{\log n}{n}\right]^{2/(d_i+8)}\Bigg) \\
&\triangleq U_{est}(q,m,t) \qquad (9)
\end{aligned}$$

where n *is the sample size and* d_i *is the dimensionality of the subspace* T_i. *Furthermore, the sum in the last equality is over at most* $\min(k, \lceil \lg m\rceil)$ *sub-spaces.*

Roughly speaking, the above result states that in estimating the m-th eigenfunction of the factorized operator A^t_q, the error is bounded by sum of the estimation errors in *at most* $\lceil \lg m\rceil$ subspaces each of which has a reduced dimensionality from d to d_i.

The main implication of the above theorems can be summarized as follows: suppose the underlying distribution p is equal or close to the factorized distribution q. If the procedure in Lemma 1 is used to estimate the principal eigenfunctions of A^t_p, the upper bound on the approximation error of these eigenfunctions will be small because p and q are close (Theorem 1). Moreover, the upper bound on the estimation error will involve only a few independent subspaces induced by q each of which has a reduced dimensionality and therefore has an exponentially faster convergence rates (Theorem 2). As a result, we get smaller total error upper bound $U_{total}(q,m,t) = U_{app}(q,m,t) + U_{est}(q,m,t)$ compared to the error bound for the standard diffusion map (Note that using the trivial partition $\boldsymbol{\mathcal{T}_1} = \{V\}$ is equivalent to the standard diffusion map).

4.3 Finding The Best Partition

So far, we have assumed that for the given problem a good partition of variables is known. This is a reasonable assumption in those problems where the (unconditional) independencies among the variables are known in advance. For instance, in object recognition problem, one may consider the edge and the texture features of the input images to be almost independent. However, in many other problems, the independencies and week dependencies among variables (and therefore the optimal partitioning) are not a priori known and need to be discovered from the data. To this end, we need an optimization criterion to evaluate the goodness of different partitions w.r.t. the task in hand. In this paper, we use the *estimated* total error for estimation of the major eigenfunctions to find a nearly optimal partition of the variables for factorized diffusion mapping. More formally, given an unlabeled dataset $\mathcal{D}$, we want to find the partition that minimizes $\mathcal{E}_{total}(q,m,t)$. However, we face the following two big challenges to solve this optimization problem. First, since we do not know the true eigenfunctions, we cannot directly compute the total error and need to estimate it. One approach to estimation of $\mathcal{E}_{total}$ is to use the upper bound U_{total} as a proxy for $\mathcal{E}_{total}$. However, the problem with this solution is we need to estimate the constants for error bounds in Theorems 1 and 2 as well as the true multiplicative eigengaps which is not easy in general for real problems; let alone the fact that these bounds are not tight anyway. To get around these problems, in our framework, we use a bootstrapping algorithm to estimate $\mathcal{E}_{total}(q,m,t)$. More precisely, from the given sample $\mathcal{D}$ of size n, we draw b bootstrap subsamples $\mathcal{D}_1,\ldots,\mathcal{D}_b$ of size $n/2$ each. Then the total error for the given partition $\boldsymbol{\mathcal{T}_k}$ is estimated as:

Algorithm 1 Greedy Partitioning

```
input: dataset $\mathcal{D}$ with features $V$
output: the optimal partitioning $\boldsymbol{\mathcal{T}^*}$
$k \leftarrow 1$, $\boldsymbol{\mathcal{T}_1} \leftarrow V$
loop
    for all $T_i \in \boldsymbol{\mathcal{T}_k}$ do
        $\{\tilde{T}_{i1}, T_i \backslash \tilde{T}_{i1}\} \leftarrow Qu(\mathcal{D}, T_i)$
        $\Delta_i \leftarrow \Delta_{total}(\{\tilde{T}_{i1}, T_i\backslash\tilde{T}_{i1}\} \mid \boldsymbol{\mathcal{T}_k})$
    end for
    $j \leftarrow \arg\max_{1\leq i\leq k} \Delta_i$
    if $\Delta_j > 0$ then
        $\boldsymbol{\mathcal{T}_k} \leftarrow \boldsymbol{\mathcal{T}_k}\backslash\{T_j\} \cup \{T_{j1}, T_{j2}\}$
        $k \leftarrow k + 1$
    else
        $\boldsymbol{\mathcal{T}^*} \leftarrow \boldsymbol{\mathcal{T}_k}$; stop
    end if
end loop
```

$$\hat{\mathcal{E}}_{total}(q,m,t) = \frac{1}{b}\sum_{i=1}^{b} \|\hat{u}^t_{p,m,\mathcal{D}} - \hat{u}^t_{q,m,\mathcal{D}_i}\|_2^2 \qquad (10)$$

Here, $\hat{u}^t_{p,m,\mathcal{D}}$ is the estimated eigenvector over the sample $\mathcal{D}$ using no partitioning whereas $\hat{u}^t_{q,m,\mathcal{D}_i}$ denotes the estimated multiplicative eigenvector over the bootstrap subsample $\mathcal{D}_i$ if the partitioning $\boldsymbol{\mathcal{T}_k}$ is applied.

Second, even after estimating the total error, we still need to find the optimal partition that minimizes the estimated error which is an NP-hard problem. To address this issue, we develop a greedy algorithm that recursively splits the variable set V into disjoint subsets and and stops when $\hat{\mathcal{E}}_{total}(q, m, t)$ cannot be decreased anymore. Let us start with the following definitions:

Definition 1. *Denoted by $\boldsymbol{\mathcal{T}'_{k+1}} \succ_i \boldsymbol{\mathcal{T}_k}$, $\boldsymbol{\mathcal{T}'_{k+1}}$ is defined to be an immediate refinement of $\boldsymbol{\mathcal{T}_k}$ on the subset $T_i \in \boldsymbol{\mathcal{T}_k}$ if*

$$\boldsymbol{\mathcal{T}'_{k+1}} = \boldsymbol{\mathcal{T}_k} \backslash \{T_i\} \cup \{T_{i1}, T_{i2}\}$$

where $T_{i1}, T_{i2} \neq \phi$, $T_{i1} \cup T_{i2} = T_i$ and $T_{i1} \cap T_{i2} = \phi$.

Definition 2. *Suppose $\boldsymbol{\mathcal{T}'_{k+1}} \succ_i \boldsymbol{\mathcal{T}_k}$ with the split $\{T_{i1}, T_i \backslash T_{i1}\}$ of T_i. The error gain of the split $\{T_{i1}, T_i \backslash T_{i1}\}$ applied on $\boldsymbol{\mathcal{T}_k}$ is defined as:*

$$\begin{aligned} &\Delta_{total}(\{T_{i1}, T_i \backslash T_{i1}\} \mid \boldsymbol{\mathcal{T}_k}) \triangleq \\ &\qquad \hat{\mathcal{E}}_{total}(q_{\boldsymbol{\mathcal{T}_k}}, m, t) - \hat{\mathcal{E}}_{total}(q_{\boldsymbol{\mathcal{T}'_{k+1}}}, m, t) \end{aligned} \tag{11}$$

Furthermore, the optimal error gain of splitting T_i in $\boldsymbol{\mathcal{T}_k}$ is defined to be:

$$\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k}) \triangleq \Delta_{total}(\{T^*_{i1}, T_i \backslash T^*_{i1}\} \mid \boldsymbol{\mathcal{T}_k}) \tag{12}$$

where

$$T^*_{i1} = \arg\max_{T_{i1} \subset T_i} \Delta_{total}(\{T_{i1}, T_i \backslash T_{i1}\} \mid \boldsymbol{\mathcal{T}_k}) \tag{13}$$

For now, suppose we can efficiently compute $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ for all $T_i \in \boldsymbol{\mathcal{T}_k}$. Then given the current partition $\boldsymbol{\mathcal{T}_k}$, the greedy algorithm picks the subset $T_i \in \boldsymbol{\mathcal{T}_k}$ with the maximum gain $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ to be split into $\{T^*_{i1}, T_i \backslash T^*_{i1}\}$ and generates $\boldsymbol{\mathcal{T}'_{k+1}}$ for the next iteration. The algorithm stops when the gains for all subsets in the current partition are negative. Of course, this algorithm is based on the assumption that $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ is efficiently computable which is not the case because of the intractable set maximization problem in Eq. (13). To address this problem, first we define the gain for the approximation error *upper bound* obtained from splitting T_i into $\{T_{i1}, T_i \backslash T_{i1}\}$ as:

$$\begin{aligned} &\Delta^U_{app}(\{T_{i1}, T_i \backslash T_{i1}\} \mid \boldsymbol{\mathcal{T}_k}) \\ &\qquad \triangleq U_{app}(q_{\boldsymbol{\mathcal{T}_k}}, m, t) - U_{app}(q_{\boldsymbol{\mathcal{T}'_{k+1}}}, m, t) \\ &\qquad = -C \cdot MI(T_{i1}, T_i \backslash T_{i1}) \end{aligned} \tag{14}$$

where $MI(X, Y)$ denotes the *mutual information* between the random vectors X and Y and C is the constant defined in Theorem 1. The equality in Eq. (14) can be obtained from the result of Theorem 1 using some basic algebra. We propose to use Δ^U_{app} instead of Δ_{total} in Eq. (13) to find the split $\{\tilde{T}_{i1}, T_i \backslash \tilde{T}_{i1}\}$ as an approximation to $\{T^*_{i1}, T_i \backslash T^*_{i1}\}$; that is,

$$\begin{aligned} \tilde{T}_{i1} &= \arg\max_{T_{i1} \subset T_i} \Delta^U_{app}(\{T_{i1}, T_i \backslash T_{i1}\} \mid \boldsymbol{\mathcal{T}_k}) \\ &= \arg\min_{T_{i1} \subset T_i} MI(T_{i1}, T_i \backslash T_{i1}) \end{aligned} \tag{15}$$

Using this heuristic, finding the best splitting inside each subset reduces to finding the most independent bi-split of the subset. The benefit of using this heuristic is that the optimization function in Eq. (15) is a symmetric submodular function which can be minimized using the Queyranne algorithm in $O(|T_i|^3)$ [16]. The disadvantage is, at the level of finding the best split inside each subset, we do not exactly maximize the estimated total error anymore. However at one level higher, when the algorithm decides which subset in the current partition should be split, it looks at the estimated total error, which is the original objective function we aim to minimize.

Once the split $\{\tilde{T}_{i1}, T_i \backslash \tilde{T}_{i1}\}$ is found, we can plug it in Eq. (12) to compute $\tilde{\Delta}_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ as an approximation to $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ for all $T_i \in \boldsymbol{\mathcal{T}_k}$. Algorithm 1 above summarizes the greedy partitioning algorithm. Note that $Qu(\mathcal{D}, T_i)$ in Algorithm 1 denotes the Queyranne algorithm which finds the splitting of T_i into $\{\tilde{T}_{i1}, T_i \backslash \tilde{T}_{i1}\}$ that minimizes Δ^U_{app}.

There are a couple of points regarding the proposed algorithm in this section to be clarified.

(1) Although the estimation error $\mathcal{E}_{est}$ is not used in finding the best splitting of each T_i, it is implicitly included in Δ^*_{total} and therefore is used to decide which T_i should be split in the next iteration.

(2) $\Delta^U_{app}(T_i \mid \boldsymbol{\mathcal{T}_k})$ only depends on the subsets T_{i1} and $T_i \backslash T_{i1}$ inside T_i and does not change if we refine other T_j's. However, this isn't true for $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$; that is, $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ depends on the whole partition $\boldsymbol{\mathcal{T}_k}$ and will change if any members of $\boldsymbol{\mathcal{T}_k}$ is split. Because of this, we cannot apply the splitting in all T_i's at the same time; in fact, any new split will change $\Delta^*_{total}(T_i \mid \boldsymbol{\mathcal{T}_k})$ for all T_i's.

(3) In practice we need a robust method to estimate the mutual information between different subsets of continuous random variables from the sample $\mathcal{D}$. One candidate is the Maximum Likelihood Density Ratio method [21] which roughly has the convergence rate of $O_p(n^{-\frac{1}{2}})$ [22].

(4) Depending on the size of problem and the method used for estimating mutual information, the optimization in Eq. (15) might be still too slow. To alleviate this problem in practice, one can substitute line 6 in Algorithm 1 with any method

that finds nearly independent partitions of variables (e.g. partitioning of the covariance matrix).

(5) One needs to decide for which eigenfunction (i.e. which m) $\hat{\mathcal{E}}_{total}(q, m, t)$ should be minimized in the greedy partitioning algorithm. In our experiments, we have used a weighted average error over the first four principal eigenfunctions.

5 Experimental Results

Synthetic Data: In the first experiment, our goal is to cluster the synthetic 3D dataset in Figure 1 (two balls surrounded by two rings) using spectral clustering. In particular, we applied K-means in the embedded spaces induced by the factorized diffusion map and the standard diffusion map. The dimension of the embedded spaces for both mappings is 3 using the first three non-trivial eigenvectors of the corresponding operators. Assuming the independence $X, Y \perp Z$ is known in advance, we passed the partition $\boldsymbol{\mathcal{T}_2} = \{\{X, Y\}, \{Z\}\}$ to the factorized diffusion mapping algorithm (induced by Lemma 1). To assess the performance of mappings, we measure the divergence of the clustering result in each case from the true cluster labels. To do so, we have used the normalized *variation of information* which is a distance metric between two clusterings [14]. This metric measures the conditional entropy of cluster labels given the true labels and vice versa (the smaller this metric is, the closer two clusterings are). Figure 2(A) shows the variation of information for the two methods with the true cluster labels as the sample size changes. We also show the performance of standard K-means without any spectral embedding (the black curve). The curves are averages over 20 repetitions with the error bars showing the 95% confidence intervals.

As the results show, for small sample sizes there is no difference between the performance of the two spectral methods. However, as we increase the sample size, our method starts to outperform the standard diffusion map leading to significantly smaller variation of information with the true cluster labels. As we continue increasing the sample size, the difference between the two methods starts decreasing with both methods eventually reaching the perfect clustering given the sample size is sufficiently large (700 for our method). According to these observations, we conclude that the extra knowledge regarding the underlying distribution of data (i.e. the independence relation) is particularly useful for mid-range sample sizes and can significantly improve the results of spectral clustering. However, for very small or very large sample sizes, this extra piece of information may not make a significant difference. Also, the standard K-means performed very poorly compared to the spectral methods.

n	k*	Baseline α	Factorized α
140	5	0.727 ± 0.007	0.755 ± 0.007
280	3	0.707 ± 0.006	0.748 ± 0.007
700	2	0.704 ± 0.005	0.764 ± 0.006

Table 1: Results of Greedy Partitioning on image data set for different sample sizes

Image Data: In the second experiment, we have applied our framework on the image segmentation dataset [1]. This dataset consists of 2310 instances. Each instance was drawn randomly from a database of seven outdoor categories. The image, a 3 × 3 region, was hand-segmented to create a classification for each region. The seven classes are brickface, sky, foliage, cement, window, path, and grass. Each of the seven categories is represented by 330 instances. The extracted features are 19 continuous attributes that describe the position of the extracted image, line densities, edges, and color values. We have treated this classification problem as a clustering task with each class regarded as one cluster. The main reason for choosing this dataset is the features conceptually seem to be divided into nearly independent subsets (e.g. the position vs. the edge features). Figure 2(B) also shows the empirical covariance matrix of this dataset with nearly blocked structure which again indicates the existence of independent feature subsets. However, there might still exist some non-linear dependencies among features and therefore we cannot completely trust on the block structure suggested by the covariance matrix as the true partitioning. This observation motivates utilizing the proposed Greedy Partitioning algorithm to automatically find the best partition for factorized diffusion mapping of the data.

Figure 2(C) shows the optimization paths of the Greedy Partitioning algorithm for different sample sizes (the plots are averaged over 10 runs). The x-axis shows the number subsets in the partitioning on the variables while the y-axis is the total error estimated using bootstrapping in the log-scale (with the minimums marked on the plots). As the figure shows, all of the plots have the same general trend: namely as we start partitioning the features, there is a significant drop on the estimated total error until the total error reaches a minimum. We attribute this behavior to the decrease in the estimation error while introducing very small approximation error. However, if we continue refining the partitioning, the increase in the approximation error will dominate the decrease in the estimation error and therefore the total error starts increasing again. It is also apparent from the plots that as the sample size increases the estimated total error decreases for a fixed number of partitions. Finally,

[1] http://archive.ics.uci.edu/ml/datasets /Image+Segmentation

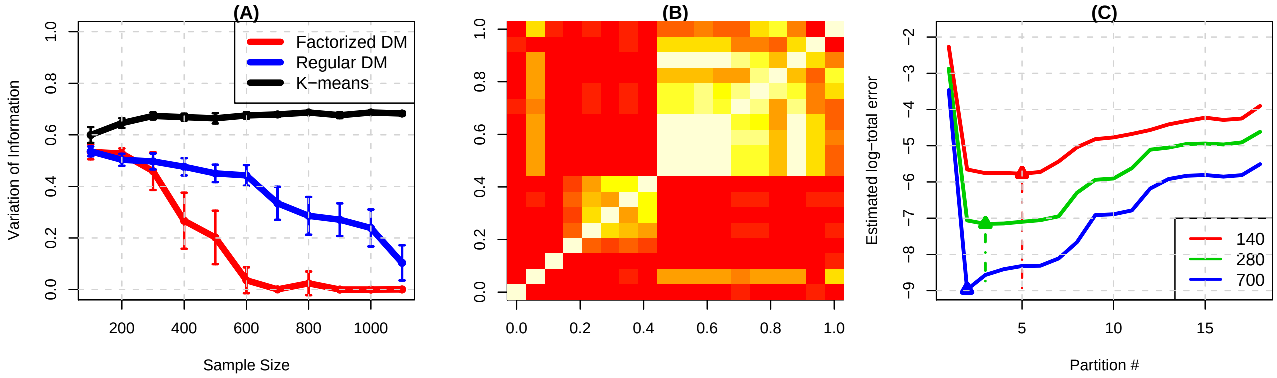

Figure 2: (A) Clustering results on synthetic data (B) Covariance matrix (C) The bootstrapping error for different partitions on the image data of features in the image data set

note that the position of minimum is shifted to the left (i.e. toward smaller partition numbers) as we increase the sample size. This observation, in fact, shows that for smaller sample sizes, the algorithm automatically regularizes more by imposing more independence assumptions (i.e. more refined partitioning) in order to get more accurate estimation of eigenfunctions.

Having found the optimal partitioning and used it for the factorized diffusion mapping, we can feed the resulted mapping to K-means to find the clusters. To evaluate the result of clustering given the true cluster labels, one option is to use the variation of information score as before. However, we observed that the results of K-means were more sensitive to initial centers in this real-world problem. To alleviate this issue, we develop a new evaluation metric called *separation* which assesses how separated the true cluster are in the embedded space, independent of the initial cluster positions for K-means. To compute this metric: given the new coordinates of data in the embedded space $\{z^{(1)}, \ldots, z^{(n)}\}$ and the true cluster labels, we compute the center μ_i for cluster C_i in the embedded space. Each C_i has n_i data points; we define w_i to be the number of data points among the n_i closest points to μ_i which actually belong to the cluster C_i (using the true labels). The separation is computed as $\alpha = \sum_i w_i/n$ which is a number in $[0, 1]$. For $\alpha = 1$, we have the perfect separation meaning that given a good set of initial points, K-means can completely separate the clusters based on the true labels. In fact, this metric is equivalent to *clustering purity metric* when K-means generates the ideal clustering by finding clusters centered at μ_i's. Table 1 summarizes the optimal number of partitions k^* found for each sample size as well as the separation α (and its 95% CI) for both standard and factorized diffusion maps. All the results are the average over 10 runs. As the results show, using factorized diffusion embedding, the separation of clusters in the embedded space is significantly improved.

6 Conclusions

In this paper, we utilized the existence of independence structure in the underlying distribution of data to estimate diffusion maps. In particular, we studied the reduction on the estimation error of diffusion eigenfunctions resulting from a factorized distribution. We showed that if the underlying space is factorized into independent subspaces, the estimation error of the major eigenfunctions can be decomposed into errors in only a subset of these subspaces each of which has a much smaller dimensionality than the original space. Since in many real problems, the factorized distribution either does not exist or is not known in advance, we studied how much bias is introduced if we impose the factorized distribution assumption. To find the optimal trade-off between the approximation bias and the estimation error, we developed a greedy algorithm for finding a factorization that minimizes the estimated total error. The experimental results showed that the factorized approximation can significantly improve the results of spectral clustering on both the synthetic and image data sets.

The fundamental intuition underlying this work is that the density estimation problem and the spectral analysis of data are closely related. Hence, the same structural assumptions that can help us to reduce the complexity of learning for density estimation purposes, can also help us for the empirical spectral analysis.

Acknowledgements

This research work was supported by grants 1R01LM010019-01A1 and 1R01GM088224-01 from the NIH and by the Predoctoral Andrew Mellon Fellowship awarded to Saeed Amizadeh for the school year 2011-2012. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[1] F. R. Bach and M. I. Jordan. Learning spectral clustering. In *Advances in Neural Information Processing Systems 16*, 2004.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002.

[3] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. In *Computational Learning Theory*, pages 486–500, 2005.

[4] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[5] F.R.K. Chung. *Spectral Graph Theory.* Amer. Math. Society, 1997.

[6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, USA, 2000.

[7] T. Cox and M. Cox. *Multidimensional Scaling.* Chapman & Hall, 1994.

[8] Rob Fergus, Yair Weiss, and Antonio Torralba. semi-supervised learning in gigantic image collection. In *NIPS*, 2009.

[9] Evarist Giné and Armelle Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Ann. Inst. Henri Poincaré (B)*, 38(6):907–921, 2002.

[10] Evarist Gine and Vladimir Koltchinskii. Empirical graph laplacian approximation of laplace–beltrami operators: Large sample results. *the IMS Lecture Notes Monograph Series by the Institute of Mathematical Statistics*, 51, December 27 2006.

[11] R. Jin, C. Ding, and F. Kang. A probabilistic approach for optimizing spectral clustering. In *Advances in Neural Information Processing Systems 18*, 2006.

[12] Stephane Lafon and Ann B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1393–1403, September 2006.

[13] Ann B. Lee and Larry Wasserman. Spectral connectivity analysis. *Journal of the American Statistical Association*, 2010.

[14] Marina Meila. Comparing clusterings by the variation of information. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 2003.

[15] Boaz Nadler, Stephane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. In *Applied and Computational Harmonic Analysis: Special issue on Diffusion Maps and Wavelets*, 2006.

[16] Mukund Narasimhan and Jeff A. Bilmes. PAC-learning bounded tree-width graphical models. In *UAI-04*, pages 410–417. AUAI Press, 2004.

[17] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2001.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[19] A. Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21:128–134, 2006.

[20] Amarnag Subramanya and Jeff Bilmes. Entropic graph regularization in non-parametric semi-supervised classification. In *NIPS*, 2009.

[21] Taiji Suzuki, Masashi Sugiyama, Jun Sese, and Takafumi Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. *Journal of Machine Learning Research - Proceedings Track*, 4:5–20, 2008.

[22] Taiji Suzuki, Masashi Sugiyama, and Toshiyuki Tanaka. Mutual information approximation via maximum likelihood estimation of density ratio. In *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 1*, pages 463–467. IEEE Press, 2009.

[23] Hamed Valizadegan, Rong Jin, and Anil K. Jain. Semi-supervised boosting for multi-class classification. In *Principles of Data Mining and Knowledge Discovery*, pages 522–537, 2008.

[24] Ulrike von Luxburg, Mikhail Belkin, and Max Planck. Consistency of spectral clustering. *Annals of Statistics*, 36:555–586, 2008.

[25] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608, 2005.

[26] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[27] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[28] Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *NIPS*, 2005.