Iterative Spectral Method for Alternative Clustering

Chieh Wu, S. Ioannidis, M. Sznaier, X. Li, D. Kaeli, J.G Dy
Northeastern University

Abstract

Traditional clustering algorithms partition of an input dataset into clusters. However, data can often be interpreted in multiple ways; alternative clustering aims to address this by finding an alternative partition, given a dataset and an existing clustering as input. One of the state-of-the-art approaches is Kernel Dimension Alternative Clustering (KDAC). KDAC simultaneously discovers complex nonlinear alternative cluster structures as well as a low-dimension subspace in which the alternative clusters reside. Although KDAC discovers high quality alternative clusters, it involves a non-convex, computationally intensive optimization, thereby lacking the speed and scalability required for human machine interactive exploratory data analysis. We propose a novel Iterative Spectral Method (ISM) that greatly improves the scalability of KDAC. Our algorithm is intuitive, relies on easily implementable spectral decompositions, and comes with theoretical guarantees. Most importantly, it is highly efficient: its computation time improves upon existing implementations of KDAC by as much as 3 orders of magnitude.

1 Introduction

Clustering, i.e., the process of grouping similar objects in a dataset together, is a classic problem. It is extensively used for exploratory data analysis. Traditional clustering algorithms typically identify a single partitioning of a given dataset. However, data is often multi-faceted and can be both interpreted and clustered through multiple viewpoints (or, *views*). For example, the same face data can be clustered based on either identity or based on pose. In real applications, partitions generated by a clustering algorithm may not correspond to the view a user is interested in.

In this paper, we address the problem of finding an *alternative clustering*, given a dataset and an existing, pre-computed clustering. Ideally, one would like the alternative clustering to be *novel* (i.e., non-redundant) w.r.t. the existing clustering to reveal a new viewpoint to the user. Simultaneously, one would like the result to reveal partitions of high clustering *quality*. Several recent papers propose algorithms for alternative clustering [1, 2, 3, 4, 5, 6]. Among them, Kernel Dimension Alternative Clustering (KDAC) is a flexible approach, shown to have superior performance compared to several competitors [6]. KDAC is as powerful as spectral clustering in discovering arbitrarily-shaped clusters (including ones that are not linearly separable) that are non-redundant w.r.t. an existing clustering. As an additional advantage, KDAC can simultaneously learn the subspace in which the alternative clustering resides.

The flexibility of KDAC comes at a price: the KDAC formulation involves optimizing a non-convex cost function constrained over the space of ortogonal matrices (i.e, the Stiefel manifold). Niu et al. [6] proposed a Dimension Growth (DG) heuristic for solving this optimization problem, which is nevertheless highly computationally intensive. We elaborate on its complexity in Section 2; experimentally, DG is quite slow, with an average convergence time of 8.4 hours on an Intel Xeon CPU, for a 1000 sample-sized synthetic data (c.f. Section 4). This limits the applicability of KDAC in interactive exploratory data analysis settings, which often require results to be presented to a user within a few seconds. It also limits the scalability of KDAC to large data. Alternately, one can solve the KDAC optimization problem by gradient descent on a Stiefel manifold (SM) [7]. However,

Submitted to 31th Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

given the lack of convexity, both DG or SM are prone to get trapped to local minima. Multiple iterations with random initializations are required to ameliorate the effect of locality. This increases computation time, and decreases also in effectiveness as the dimensionality of the data increases: the increase in dimension rapidly expands the search space and the abundance of local minima. As such, with both DG and SM, the clustering quality is negatively effected by an increase in dimension.

Our Contributions. Motivated by the above issues, we make the following contributions:

- We propose an Iterative Spectral Method (ISM), a *novel algorithm* for solving the non-convex optimization constrained on a Stiefel manifold problem inherent in KDAC. Our algorithm has several highly desirable properties. First, it *significantly outperforms* traditional methods such as DG and SM in terms of both computation time and quality of the produced alternative clustering. Second, the algorithm relies on an *intuitive use of iterative spectral decompositions*, making it both easy to understand as well as easy to implement, using off-the-shelf libraries.
- ISM has a natural initialization, constructed through a Taylor approximation of the problem's Lagrangian. Therefore, high quality results can be obtained without random restarts in search of a better initialization. We show that this initialization is a contribution in its own right, as its use improves performance of competitive algorithms.
- We provide a theoretical analysis of ISM, establishing *theoretical guarantees* on its fixed point. In particular, we establish conditions under which the fixed point of ISM satisfies both the 1st and 2nd order necessary conditions for local optimality.
- We extensively evaluate the performance of ISM in solving KDAC with synthetic and real data under various clustering quality and cost measures. Our results show an improvement in execution time by up to a factor of 110 and 5848, compared to SM and DG, respectively. At the same time, ISM outperforms SM and DG in all clustering quality measures along with a lower computational cost. Our analysis of its scalability indicates a divergence in execution time between ISM and the competing algorithms, with ISM showing the best scalability.

Related Work. There exist two general modes of discovering alternative clusterings – simultaneously or iteratively. Simulataneous approaches find the multiple alternative clusterings at the same time [8, 9, 3, 10, 11, 12, 13]. Iterative approaches find an alternative clustering given existing clustering [2]. Since this work focuses on the iterative paradigm, we elaborate on the related work along these lines. Alternative clustering methods differ in how they measure novelty and cluster quality. Gondek and Hofmann [1] find an alternative clustering by conditional information (CI) bottleneck. Bae and Bailey [14] perform agglomerative clustering with cannot-link constraints imposed on the data points that belong together in the existing clustering. Cui et al. [5] find an alternative clustering by projecting the data to a subspace orthogonal to the existing clustering. Qi and Davidson [15] search for novelty by minimizing the Kullback-Leiber (KL) divergence between the original data and the transformed data subject to the constraint that the sum-squared-error between samples in the projected space with the existing clusters is small. Dang and Bailey [16] find quality clusters by maximizing the mutual information (MI) between the alternative clusters and the data while simultaneously ensuring novelty by minimizing the MI between alternative and existing clusterings.

KDAC's ability to detect arbitrarily-shaped clusters is due to its use the Hilbert-Schmidt Independence Criterion (HSIC) [17] as a cluster quality measure. HSIC is motivated by the objective function of spectral clustering. Moreover, since HSIC models non-linear dependence, it is also utilized by KDAC to measure novelty. In contrast, e.g., the orthogonal subspace projection approach in [5] is limited, as it only captures linear dependencies. Other approaches, such as [1, 16], can take non-linear dependencies into account by utilizing information theoretic measures. However, doing so requires estimating joint probability distributions. The advantage of KDAC over such approaches is that it utilizes HSIC for measuring novelty and cluster quality, which can capture non-linear dependencies through kernels, without having to explicitly learn the joint probability distributions; empirically, it significantly outperforms aforementioned schemes in clustering quality [6].

2 Kernel Dimension Alternative Clustering (KDAC)

In alternative clustering, a dataset is provided along with existing clustering labels. Given this as input, we seek a *new* clustering that is (a) distinct from the existing clustering, and (b) has high quality with respect to a clustering quality measure. An example illustrating this is shown in Figure 1. This dataset comprises 400 points in $\mathbb{R}^4$. Projected to the first two dimensions, the dataset contains

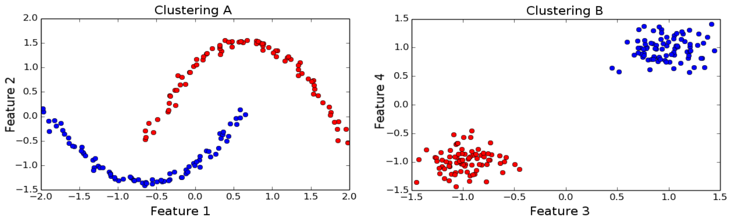

Figure 1: Four-dimensional moon dataset. Projection into the first two dimensions reveals different clusters than projection to the latter two dimensions.

two clusters of intertwining parabolas, shown as Clustering A. Projected to the last two dimensions, the dataset contains two Gaussian clusters, shown as Clustering B. Points clustered together in one view can be in different clusters in the alternative view. In alternative clustering, given (a) the dataset, and (b) one of the two possible clusterings (e.g., Clustering B), we wish to discover the alternative clustering illustrated by the different view.

Formally, let $X \in \mathbb{R}^{n\times d}$ be a dataset with n samples and d features, along with an existing clustering $Y \in \mathbb{R}^{n\times k}$, where k is the number of clusters. If x_i belongs to cluster j, then $Y_{i,j} = 1$; otherwise, $Y_{i,j} = 0$. We wish to discover an alternative clustering $U \in \mathbb{R}^{n\times k}$ on some lower dimensional subspace of dimension $q \ll d$. Let $W \in \mathbb{R}^{d\times q}$ be a projection matrix such that $XW \in \mathbb{R}^{n\times q}$. We seek the optimal projection W and clustering matrix U that maximizes the statistical dependence between XW with U, yielding a high clustering quality, while minimizing the dependence between XW and Y, ensuring the novelty of the new clustering. Denoting DM as a Dependence Measure function, and using λ as a weighing constant, this optimization can be written as:

$$\text{Maximize:}\quad \mathrm{DM}(XW, U) - \lambda\,\mathrm{DM}(XW, Y), \tag{1a}$$

$$\text{subject to:}\quad W^T W = I, U^T U = I. \tag{1b}$$

As in spectral clustering, the labels of the alternative clustering are retrieved by performing K-means on matrix U, treating its rows as samples. There are many potential choices for DM. The most well-known measures are correlation and mutual information (MI). While correlation performs well in many applications, it lacks the ability to measure non-linear relationships. Although there is clear relationship in Clustering A in Figure 1, correlation would mistakenly yield a value of nearly 0 . As a dependence measure, MI is superior in that it also measures non-linear relationships. However, due to the probabilistic nature of its formulation, a joint distribution is required. Depending on the distribution, the computation of MI can be prohibitive.

For these reasons, the Hilbert Schmidt Independence Criterion (HSIC) [17] has been proposed for KDAC [6]. Like MI, it captures non-linear relationships. Unlike MI, HSIC does not require estimating a joint distribution, and it relaxes the need to discretize continuous variables. In addition, as shown by Niu et al. [6], HSIC is mathematically equivalent to spectral clustering, further implying that a high HSIC between the data and U yields high clustering quality. A visual comparison of HSIC and correlation can be found in Figure 5 of Appendix G in the supplement.

Using HSIC as a dependence measure, the objective of KDAC becomes

$$\text{Maximize:}\quad \mathrm{HSIC}(XW, U) - \lambda\,\mathrm{HSIC}(XW, Y), \tag{2a}$$

$$\text{subject to:}\quad W^T W = I, U^T U = I. \tag{2b}$$

where $\mathrm{HSIC}(X, Y) \equiv \frac{1}{(n-1)^2}\,\mathrm{Tr}(K_X H K_Y H)$. Here, the variables K_X and K_Y are Gram matrices, and the H matrix is a centering matrix where $H = I - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$ with $\mathbb{1}$ the n-sized vector of all ones. The elements of K_X and K_Y are calculated by kernel functions $k_X(x_i, x_j)$ and $k_Y(y_i, y_j)$. The kernel functions for Y and U used in KDAC are $K_Y = YY^T$ and $K_U = UU^T$, and the kernel function for XW is the Gaussian $k_{XW}(x_i, x_j) = \exp(-\mathrm{Tr}[(x_i - x_j)^T W W^T (x_i - x_j)]/(2\sigma^2))$. Due to the equivalence of HSIC and spectral clustering, the practice of normalizing the kernel K_{XW} is adopted from spectral clustering by Niu et al. [6] That is, for K_{XW} the unnormalized Gram matrix, the normalized matrix is defined as $D^{-1/2} K_{XW} D^{-1/2}$ where $D = \mathrm{diag}(\mathbb{1}_n^T K_{XW})$ is a diagonal matrix whose elements are the column-sums of K_{XW}.

Algorithm 1: KDAC Algorithm

```
Input   : dataset $X$, original clustering $Y$
Output  : alternative clustering $Y$
Initialize $W_0$ using $W_{\text{init}}$ from (12)
Initialize $U_0$ from original clustering
Initialize $D_0$ from $W$ and original clustering
while (U not converged) or (W not converged) do
    Update $D$
    Update $W$ by solving Equation (4)
    Update $U$ by solving Equation (3)
Clustering Result ← Apply Kmeans to $U$
```

Algorithm 2: ISM Algorithm

```
Input   : $U$,$D$,$X$, $Y$
Output  : $W^*$
Initialize $W_0$ to the previous value of $W$ in the
  master loop of KDAC.
while $W$ not converged do
    $W \leftarrow \text{eig}_{\text{min}}(\Phi(W))$;
```

2.1 KDAC Algorithm

The optimization problem (2) is non-convex. The KDAC algorithm solves (2) using alternate maximization between the variables U, W and D, updating each while holding the other two fixed. After convergence, motivated by spectral clustering, U is discretized via K-means to provide the alternative clustering. The algorithm proceeds in an iterative fashion, summarized in Algorithm 1. In each iteration, variables D, U, and W are updated as follows:

Updating D: While holding U and W constant, D is computed as $D = \text{diag}(\mathbb{1}_n^T K_{XW})$. Matrix D is subsequently treated as a scaling constant throughout the rest of the iteration.

Updating U: Holding W and D constant and solving for U, (2) reduces to :

$$\max_{U:U^TU=I} \text{Tr}(U^T \mathcal{Q} U), \tag{3}$$

where $\mathcal{Q} = HD^{-1/2}K_{XW}D^{-1/2}H$. This is precisely spectral clustering [18]: (3) can be solved by setting U's columns to the k most dominant eigenvectors of $\mathcal{Q}$, which can be done in $O(n^3)$ time.

Updating W: While holding U and D constant to solve for W, (2) reduces to:

$$\text{Minimize:} \quad F(W) = -\sum_{i,j} \gamma_{i,j} e^{-\frac{\text{Tr}[W^T A_{i,j} W]}{2\sigma^2}} \tag{4a}$$

$$\text{subject to:} \quad W^T W = I \tag{4b}$$

where $\gamma_{i,j}$ are the elements of matrix $\gamma = D^{-1/2}H(UU^T - \lambda YY^T)HD^{-1/2}$, and $A_{i,j} = (x_i - x_j)(x_i - x_j)^T$ (see Appendix A in the supplement for the derivation). This objective, along with a Stiefel Manifold constraint, $W^TW = I$, pose a challenging optimization problem as neither is convex. Niu et al. [6] propose solving (4) through an algorithm termed Dimensional Growth (DG). This algorithm solves for W by computing individual columns of W separately through gradient descent (GD). Given a set of computed columns, the next column is computed by GD projected to a subspace orthogonal to the span of computed set. Since DG is based on GD, the computational complexity is dominated by computing the gradient of (4a). The latter is given by:

$$\nabla F(W) = \sum_i^n \sum_j^n \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\text{Tr}[W^T A_{i,j} W]}{2\sigma^2}} A_{i,j} W. \tag{5}$$

The complexity of DG is $O(tn^2d^2q)$, where n, d are the dataset size and dimension, respectively, q is the dimension of the subspace of the alternative clustering, and t is the number of iterations of gradient descent. The quadratic nature of the cost to both sample size and dimension is consistent with our experiments (Sec. 4), as DG's performance sharply deteriorates as either increase. An alternative approach to optimize (4) is through classic methods for performing optimization on the Stiefel Manifold [7]. The computational complexity of this algorithm is dominated by the computation of the gradient and a matrix inversion. This yields $O(n^2d^2q + d^3)$ complexity per step of descent on the manifold. Finally, as gradient methods applied to a non-convex objective, both SM and DG require multiple executions from random initialization points to find improved local minima. This approach becomes less effective as the dimension d increases.

3 An Iterative Spectral Method

The computation of KDAC is dominated by the W updates in Algorithm 1. Instead of using DG or SM to solve the optimization problem for W in KDAC, we propose an Iterative Spectral Method

(ISM). Our algorithm is motivated from the following observations. The Lagrangian of (4) is:

$$\mathcal{L}(W, \Lambda) = -\sum_{i,j} \gamma_{i,j} \exp\left(-\frac{\mathrm{Tr}(W^T A_{i,j} W)}{2\sigma^2}\right) - \frac{1}{2}\mathrm{Tr}(\Lambda(W^T W - I)) \tag{6}$$

Setting $\nabla_W \mathcal{L}(W, \Lambda) = 0$ gives us the equation:

$$\Phi(W)W = W\Lambda, \tag{7}$$

where

$$\Phi(W) = \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} \exp(-\frac{\mathrm{Tr}[W^T A_{i,j} W]}{2\sigma^2})A_{i,j}. \tag{8}$$

Recall that feasible W, satisfying (4b), are orthonormal. Suppose that Λ in (7) is a diagonal matrix. Then, by (7), a stationary point W of the Lagrangian (6) comprises some eigenvectors of $\Phi(W)$ as columns. Motivated by this observation, ISM attempts to find such a W in the following iterative fashion. Let W_0 be an initial matrix. Given W_k at iteration k, the matrix W_{k+1} is computed as:

$$W_{k+1} = \mathrm{eig}_{\min}(\Phi(W_k)), \quad k = 0, 1, 2, \ldots,$$

where the operator $\mathrm{eig}_{\min}(A)$ returns a matrix whose columns are the q eigenvectors corresponding to the smallest eigenvalues of A.

ISM is summarized in Alg. 2. Several important observations are in order. First, the algorithm ensures that W_k, for $k \geq 1$, is feasible, by construction: selected eigenvectors are orthonormal and satisfy (4b). Second, it is also easy to see that a fixed point of the algorithm will also be a stationary point of the Lagrangian (6) (see also Lemma 3). Though it is harder to prove, selecting eigenvectors corresponding to the *smallest* eigenvalues is key: we show that this is precisely the property that relates a fixed point of the algorithm to the local minimum conditions (see Thm. 1). Finally, ISM has several computational advantages. A single iteration involves the calculation of $\Phi(W)$, and the ensuing eigendecomposition. The latter is simple and easy to compute with standard libraries. This yields a complexity of $O(n^2d^2 + d^3)$; the cubic term can be further lowered by using methods tailored to computing the bottom eigenvectors, rather than the full eigendecomposition. Even so, for $n \gg d$, comparing this result to $O(tn^2d^2q)$ and $O(n^2d^2q)$ from DG and SM, ISM is faster than DG by a factor of tq and faster than SM by a factor of q. In practice, we observe far higher speedups ($\sim 10^3$) compared to both methods, combined with equal or better clustering quality (see Sec. 4).

Convergence Guarantees. As mentioned above, the selection of the eigenvectors corresponding to the *smallest* eigenvalues of $\Phi(W_k)$ is crucial for establishing that ISM discovers a stationary point of the Lagrangian that has good quality. In particular, we establish the following result:

Theorem 1. *For large enough σ, a fixed point W^* of Algorithm 2 satisfies the necessary conditions of a local minimum of* (4) *if $\Phi(W^*)$ is full rank.*

Proof. Our proof is organized into a series of lemmas, whose proofs are in the supplement. Our first auxiliary lemma (from [19]), establishes conditions necessary for a stationary point of the Lagrangian to constitute local minimum.

Lemma 1. *[Nocedal,Write, Theorem 12.5 [19]] (2nd Order Necessary Conditions)*

Consider the optimization problem: $\min_{W:h(W)=0} f(W)$, where $f : \mathbb{R}^{d\times q} \to \mathbb{R}$ and $h : \mathbb{R}^{d\times q} \to \mathbb{R}^{q\times q}$ are twice continuously differentiable. Let $\mathcal{L}$ be the Lagrangian of this optimization problem. Then, a local minimum must satisfy the following conditions:

$$\begin{aligned} \nabla_W \mathcal{L}(W^*, \Lambda^*) &= 0, \quad (9) \\ \nabla_\Lambda \mathcal{L}(W^*, \Lambda^*) &= 0, \quad (10) \end{aligned} \qquad \textit{and} \qquad \begin{aligned} &\mathrm{Tr}(Z^T \nabla^2_{WW} \mathcal{L}(W^*, \Lambda^*) Z) \geq 0 \\ &\quad \text{for all } Z \neq 0, \text{with } \nabla h(W^*)^T Z = 0. \end{aligned} \tag{11}$$

Armed with this result, we next characterize the properties of a fixed point of Algorithm 2:

Lemma 2. *Let W^* be a fixed point of Algorithm 2. Then it satisfies: $\Phi(W^*)W^* = W^*\Lambda^*$, where $\Lambda^* \in \mathbb{R}^{q\times q}$ is a diagonal matrix containing the q smallest eigenvalues of $\Phi(W^*)$ and ${W^*}^T W^* = I$.*

The proof can be found in Appendix B. Our next result, whose proof is in Appendix C, states that a fixed point satisfies the 1st order conditions of Lemma 1.

Lemma 3. *If W^* is a fixed point and Λ^* is as defined in Lemma 2, then W^*, Λ^* satisfy the 1st order conditions (9)(10) of Lemma 1.*

Our last lemma, whose proof is in Appendix D, establishes that a fixed point satisfies the 2nd order conditions of Lemma 1, for large enough σ.

Lemma 4. *If W^* is a fixed point, Λ^* is as defined in Lemma 2, and $\Phi(W^*)$ is full rank, then given a large enough σ, W^* and Λ^* satisfy the 2nd order condition (11) of Lemma 1.*

Thm. 1 therefore follows. □

Initialization via Taylor Approximation. An additional advantage of ISM is that it admits a natural initialization. This initialization turns out to be very important in practice, and leads to alternative clusterings of high quality (see Sec. 4). In short, consider the following approximation of the Lagrangian, obtained by replacing objective F, given by (4a), by its 2nd order Taylor approximation at $W = 0$:

$$\tilde{\mathcal{L}}(W, \Lambda) \approx -\sum_{i,j} \gamma_{i,j} \left(1 - \frac{\operatorname{Tr}(W^T A_{i,j} W)}{2\sigma^2}\right) + \frac{1}{2} \operatorname{Tr}(\Lambda(I - W^T W)).$$

Setting $\nabla_W \tilde{\mathcal{L}}(W, \Lambda) = 0$ gives $\left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}\right] W = W\Lambda$. Hence, the 2nd order Taylor approximation motivates initializing W as

$$W_{\text{init}} = \operatorname{eig}_{\min}(\textstyle\sum_{i,j} \gamma_{i,j} A_{i,j}/\sigma^2) \tag{12}$$

We use this initialization in the first master iteration of KDAC. In subsequent master iterations, W_0 (the starting point of ISM) is set to be the last value to which ISM converged to previously (i.e., to the last value used in the previous master iteration of KDAC).

4 Experimental Results

We experimentally validate the performance of ISM in terms of both speed and clustering quality. Because [6] has already performed extensive comparisons of KDAC against other alternative clustering methods, this section will concentrate on comparing ISM to competing models for optimizing KDAC: Dimensional Growth (DG) [6] and gradient descent on the Stiefel Manifold (SM) [7]. SM is the traditional approach, while DG is the approach originally proposed to solve KDAC.

Datasets. We perform experiments on four synthetic and three real data. Gauss A contains four Gaussian clusters with 40 samples and two features. Gauss B contains four Gaussian clusters with 1000 samples and four dimensions. The Gaussian clusters are rotated by 45 degrees to reside in 3D, and the fourth dimension is generated from a uniform noise distribution. This dataset tests the response of KDAC within a noisy environment. We generate two additional synthetic datasets: Moon and Moon+Noise (Moon+N). Both datasets have the first two dimensions as two parabolas and the second two dimensions as Gaussian clusters. The moon dataset with noise further includes three noise dimensions generated from a uniform distribution with 1000 samples. Since the Gaussian clusters have a compact structure and the parabolas have a non-linear structure, these datasets demonstrate KDAC's ability to handle mixed clustering structures in a uniformly noisy environment. Due to the novelty of alternative clustering, there has been very few public repository data that have at least two alternative labels. The two traditional benchmark datasets are CMU's WebKB dataset [20] and face images from the UCI KDD repository [21]. CMU's WebKB dataset consists of 1041 html pages from 4 universities. One labelling is based on universities, and an alternative labelling based on topic (course, faculty, project, student). After preprocessing by removing rare and stop words, we are left with 500 words. The face dataset consists of 640 images from 20 people in four different poses. Each image has 32x30 pixels. Images are vectorized and PCA is then used to further condense the dataset by keeping 85% of the variance, resulting in a dataset of 624 samples and 27 features. We set the existing clustering based on identity, and seek an alternative clustering based on pose. The last real dataset is The Flower image by Alain Nicolas [22], a 350x256 pixel image. The RGB values of each pixel is taken as a single sample, with repeated samples removed. This results in a dataset of 256 samples and 3 features. Although this dataset does not have labels, the quality of the alternative clustering can be visually observed. With the exception of the high dimensional WebKB data, the remaining datasets are visualized in Figure 2 (b).

(a) Clustering Quality and Execution Time Results from all Experiments

Gauss A	Best ↑	Avg NMI ↑	Avg N ↓	CQ ↑	Avg Cost ↓	Avg Time (s) ↓
ISM	**1**	**1±0**	**0±0**	**1.046±0**	**-0.938±0**	**0.012±0**
SM	**1**	**1±0**	**0±0**	**1.046±0**	**-0.938±0**	0.452±0.002
DG	**1**	**1±0**	**0±0**	**1.046±0**	19.06±0	7.829±0.028
Gauss B						
ISM	**1**	**1±0**	**0±0**	**0.76±0**	**-0.76±0**	**1.3±0**
SM	**1**	0.5±0.5	0.001 ±0.001	0.67±0.1	-0.63±0.1	2.3±0.86
DG	0.51	0.07±0.16	0.007±0.02	0.1±0.1	33±39	18059±2634
Moon						
ISM	**1**	**1±0**	**0±0**	**1.99±0**	**1.24±0**	**0.169±0**
SM	**1**	0.531±0.3	0.002±0.003	1.99±0.004	16.878±5.7	1.03±0.2
DG	**1**	0.095±0.29	0.76±0.38	0.77±0.48	39.605±22	409±252
Moon+N						
ISM	**1**	**1±0**	**0±0**	**2±0**	**-1.83±0**	**5.2±0**
SM	0.6	0.47±0.13	**0±0**	1.97±0.02	-1.35±0.12	40.1±20.6
DG	0.63	0.27±0.25	0.22±0.36	0.9±0.46	4.6±3.1	30411±1757
Flower						
ISM	-	-	-	**1.98±0**	**30.72±0**	**1.6±0**
SM	-	-	-	1.977±0.007	38.6±19	4±4
DG	-	-	-	1.47±0.36	100±44	299±58
Face						
ISM	**0.442**	**0.442±0**	**0.023±0**	**0.076±0**	**-0.074±0**	**37±0**
SM	0.339	0.263±0.05	0.036±0.01	**0.076±0**	-0.072±0.001	496±103
DG	0.051	0.025±0.02	0.388±0.08	0.025±0.007	0.137±0.06	22841±7653
Web KB						
ISM	**0.37**	**0.37±0**	**0±0**	**0.3±0**	**-0.27±0**	**140±0**
SM	0.06	0.04±0.01	0.009±0.04	0.05±0.04	0.08±0.04	15404±187
DG	0.07	0.04±0.01	0.085±0.04	0.05±0.01	0.13±0.03	94682±625

(b) Figures of all displayable Experiments.

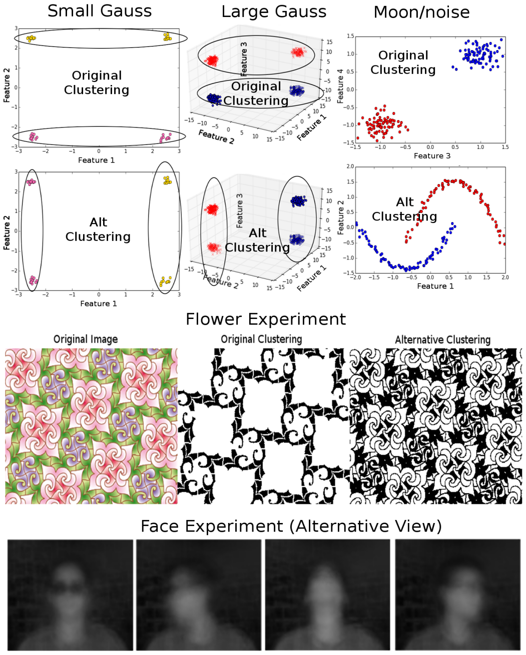

Figure 2: (a) Results from all experiments. ISM demonstrated superior clustering quality with less execution time. (b) Figures of experimental results. All original and alternative clusterings are displayed, with the exception of the Face dataset. It is observable that the original and alternative clusters are all visually obvious clusters and provide alternative views.

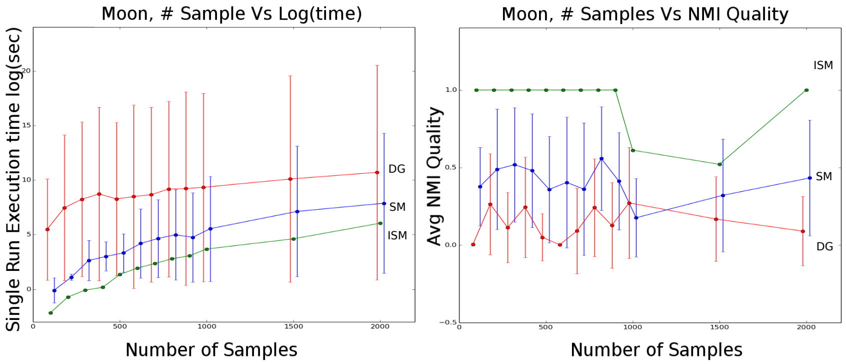

Figure 3: Number of Samples versus execution time of Moon Experiment

Evaluation Method. To measure clustering quality, both external and internal criteria have been used. External criteria measure how much the alternative labels deviate from an externally known ground truth; internal criteria measure quality based on the characteristic of the data and the partition result. For external criteria, normalized mutual information (NMI) as suggested in [23] is used. Let U and L be the alternative clustering result and the ground-truth label respectively, the NMI is defined as: $NMI(L,U) = \frac{I(L,U)}{\sqrt{H(L)H(U)}}$, where $I(L,U)$ is the mutual information between L and U, and $H(L)$ and $H(U)$ are the entropies of L and U respectively. The NMI is a measure between 0 to 1 with 0 denoting no relationship and 1 as maximum relationship. By calculating the NMI between the ground truth and the alternative labels, a measure for cluster quality can be computed. Besides comparing the alternative labels against the ground truth, the NMI of the alternative labels with the original labels is also calculated to measure novelty. In this case, the NMI is ideally low since we wish to discover clusters that are different from the original clusters. Our first internal criterion is clustering quality measured by $HSIC(XW, U)$, which is equivalent to the spectral clustering objective. High values denote high clustering quality. Another internal measure is the value of the KDAC objective function, or Equation (2). Since the objective is in the form of a cost function, a lower cost is desirable.

Besides the quality measures, we report the execution time in seconds on an Intel Xeon CPU. To avoid being trapped in a local minimum, both SM and and DG performed 10 random initialization restarts. The best and the average results along with the standard deviation of each experiment is provided. Since ISM uses a fixed initialization to immediately achieve high quality results, only one run is necessary, and the standard deviation is zero.

Results. The table in Figure 2 (a) reports the results of our experiment. The best and the average clustering quality along with its standard deviation in terms of NMI is listed in the first 2 columns. The average novelty measure (Avg N) is listed in column 3. In terms of internal criteria, the clustering quality (CQ) and the average cost function (Avg Cost) are listed in column four and five. The last column reports the average execution time required for a single execution in seconds. For all of these measures, $\pm$ one standard deviation is added to denote the variability of the results within 10 runs. The optimal direction of each measure is denoted by the $\uparrow\downarrow$, with $\uparrow$ denoting a preference towards larger values and $\downarrow$ otherwise. For each field, the optimal result is printed in bold font.

It can be observed that ISM surpasses both DG and SM in all fields. ISM consistently outperforms competing methods in terms of cluster quality, novelty and speed. This differential is especially prominent against DG, which was originally proposed for KDAC. Comparing the execution time, the maximum improvement of ISM is 5848 times against DG in the Moon+N experiment. Having a predefined initial point, ISM avoids the need to re-run the algorithm with random initializations. Techniques that require a search for the ideal starting point decrease in performance as the dimensionality of the search space increase. For this reason, the average performance for DG and SM is especially low for higher dimensional datasets. On the other hand, ISM is significantly less effected by the increase in dimension.

To assess ISM's scalability, we generate synthetic data of variable size for the moon dataset to compare the speed difference as the number of samples increases. The average time and standard deviation in seconds ($\log$ scale) as they vary with sample size is displayed on the left and the respective quality in terms of NMI with the true labels is shown on the right in Figure 3. This confirms that ISM scales best compared to DG and SM. It also still consistently performs the best in terms of cluster quality as the number of samples is varied. Note that the experiments here were run using an off the shelf Python Numpy function for eigendecomposition. There are several more scalable existing techniques for eigendecomposition. As a future extension, since ISM is a spectral method requiring only a few eigenvalues, one may use techniques such as the power method [24] to improve scalability.

5 Conclusions

We have proposed a new iterative spectral optimization technique for alternative clustering. Crucial to our construction is the stationarity condition of the Lagrangian (7). We believe our methodology and guarantees can be extended to other optimization problems involving Gaussian-kernel like objectives optimized over the Stiefel manifold. Understanding whether ISM can be applied to such objectives and be leveraged to solve broader classes of problems is a natural future direction for this work.

References

[1] David Gondek and Thomas Hofmann. Non-redundant data clustering. *Knowledge and Information Systems*, 12(1):1–24, 2007.

[2] Ying Cui, Xiaoli Z Fern, and Jennifer G Dy. Learning multiple nonredundant clusterings. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(3):15, 2010.

[3] Xuan Hong Dang and James Bailey. Generation of alternative clusterings using the cami approach. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 118–129. SIAM, 2010.

[4] Ian Davidson and Zijie Qi. Finding alternative clusterings using constraints. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 773–778. IEEE, 2008.

[5] Ying Cui, Xiaoli Z Fern, and Jennifer G Dy. Non-redundant multi-view clustering via orthogonalization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 133–142. IEEE, 2007.

[6] Donglin Niu, Jennifer G Dy, and Michael I Jordan. Iterative discovery of multiple alternativeclustering views. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1340–1353, 2014.

[7] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[8] Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. Meta clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 107–118. IEEE, 2006.

[9] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[10] Sajib Dasgupta and Vincent Ng. Mining clustering dimensions. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 263–270, 2010.

[11] Vikash K Mansinghka, Eric Jonas, Cap Petschulat, Beau Cronin, Patrick Shafto, and Joshua B Tenenbaum. Cross-categorization: A method for discovering multiple overlapping clusterings. In *Nonparametric Bayes Workshop at NIPS*, 2009.

[12] Donglin Niu, Jennifer Dy, and Zoubin Ghahramani. A nonparametric bayesian model for multiple clustering with overlapping feature views. In *Artificial Intelligence and Statistics*, pages 814–822, 2012.

[13] Leonard Poon, Nevin L Zhang, Tao Chen, and Yi Wang. Variable selection in model-based clustering: To do or to facilitate. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 887–894, 2010.

[14] Eric Bae and James Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 53–62. IEEE, 2006.

[15] ZiJie Qi and Ian Davidson. A principled and flexible framework for finding alternative clusterings. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726. ACM, 2009.

[16] Xuan-Hong Dang and James Bailey. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 573–582. ACM, 2010.

[17] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[18] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[19] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35:67–68, 1999.

[20] CMU CMU. universities webkb data, 1997. 4.

[21] Stephen D Bay, Dennis Kibler, Michael J Pazzani, and Padhraic Smyth. The uci kdd archive of large data sets for data mining research and experimentation. *ACM SIGKDD Explorations Newsletter*, 2(2):81–85, 2000.

[22] Particles of tessellations. `http://en.tessellations-nicolas.com/`. Accessed: 2017-04-25.

[23] A Strehl and J Chosh. Knowledge reuse framework for combining multiple partitions. *Journal of Machine learning Research*, 33(3):583–617.

[24] Peter Richtárik. Generalized power method for sparse principal component analysis.

Appendix A Derivation for Equation 4

Given the objective function,

$$\begin{array}{ll} \max & \mathrm{HSIC}(XW,U) - \lambda\,\mathrm{HSIC}(XW,Y) \\ U,W & \\ s.t & W^TW = I, U^TU = I. \end{array}$$

Using the HSIC measure defined, the objective function can be rewritten as

$$\begin{aligned} \mathrm{HSIC}(XW,U) - \lambda\,\mathrm{HSIC}(XW,Y) &= \mathrm{Tr}(HUU^THK_{XW}) - \lambda\,\mathrm{Tr}(HYY^THK_{XW}) \\ &= \mathrm{Tr}(H(UU^T - \lambda YY^T)HK_{XW}) \\ &= \mathrm{Tr}(\gamma K_{XW}) \\ &= \textstyle\sum_{i,j} \gamma_{i,j} K_{X_{i,j}}. \end{aligned}$$

where γ is a symmetric matrix and $\gamma = H(UU^T - \lambda YY^T)H$. By substituting the Gaussian kernel for $K_{X_{i,j}}$, the objective function becomes

$$\min_{W} \quad -\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{Tr}[W^TA_{i,j}W]}{2\sigma^2}} \qquad s.t \quad W^TW = I.$$

Appendix B Proof for Lemma 2

Proof. Algorithm 2 sets the smallest q eigenvectors of $\Phi(W_k)$ as W_{k+1}. Since a fixed point W^* is reached when $W_k = W_{k+1}$, therefore W^* consists of the smallest eigenvectors of $\Phi(W^*)$ and Λ^* corresponds with a diagonal matrix of eigenvavlues. Since the eigenvectors of $\Phi(W^*)$ are orthonormal , ${W^*}^T W^* = I$ is also satisfied. □

Appendix C Proof for Lemma 3

Proof. Using Equation (4) as the objective function, the corresponding Lagrangian and its gradient is written as

$$\mathcal{L}(W,\Lambda) = -\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{Tr}(W^TA_{i,j}W)}{2\sigma^2}} - \frac{1}{2}\,\mathrm{Tr}(\Lambda(W^TW - I)), \tag{13}$$

and

$$\nabla_W \mathcal{L}(W,\Lambda) = \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}(W^TA_{i,j}W)}{2\sigma^2}} A_{i,j}W - W\Lambda. \tag{14}$$

By setting the gradient of the Lagrangian to zero, and using the definition of $\Phi(W)$ from Equation (8), Equation (14) can be written as

$$\Phi(W)W = W\Lambda. \tag{15}$$

The gradient with respect to Λ is

$$\nabla_\Lambda \mathcal{L}(W,\Lambda) = W^TW - I. \tag{16}$$

Setting this gradient of the Lagrangian also to zero, condition (10) is equivalent to

$$W^TW = I. \tag{17}$$

By Lemma 2, a fixed point W^* and its corresponding Λ^* satisfy (15) and (17), and the lemma follows. □

Appendix D Proof for Lemma 4

The proof for Lemma 4 relies on the following three sublemmas. The first two sublemmas demonstrate how the 2nd order conditions can be rewritten into a simpler form. With the simpler form, the third lemma demonstrates how the 2nd order conditions of a local minimum are satisfied given a large enough σ.

Lemma 4.1. *Let the directional derivative in the direction of Z be defined as*

$$\mathcal{D}f(W)[Z] := \lim_{t \to 0} \frac{f(W+tZ) - f(W)}{t}. \tag{18}$$

Then the 2nd order condition of Lemma 4 can be written as

$$\operatorname{Tr}(Z^T \mathcal{D}\nabla\mathcal{L}[Z]) = \left\{ \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\operatorname{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} \left[\operatorname{Tr}(Z^T A_{i,j} Z) - \frac{1}{\sigma^2} \operatorname{Tr}(Z^T A_{i,j} W^*)^2 \right] \right\} - \operatorname{Tr}(Z^T Z \Lambda^*), \tag{19}$$

for all Z such that

$$Z^T W^* + W^{*T} Z = 0. \tag{20}$$

Proof. Observe first that

$$\nabla^2_{W^* W^*} \mathcal{L}(W^*, \Lambda^*) Z = \mathcal{D}\nabla\mathcal{L}[Z], \tag{21}$$

where the directional derivative of the gradient $\mathcal{D}\nabla\mathcal{L}[Z]$ is given by

$$\mathcal{D}\nabla\mathcal{L}[Z] = \lim_{t \to 0} \frac{\partial}{\partial t} \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\operatorname{Tr}((W^* + tZ)^T A_{i,j} (W^* + tZ))}{2\sigma^2}} A_{i,j}(W^* + tZ) - (W^* + tZ)\Lambda.$$

This can be written as

$$\mathcal{D}\nabla\mathcal{L}[Z] = T_1 + T_2 - T_3,$$

where

$$\begin{aligned}
T_1 = & \lim_{t \to 0} \frac{\partial}{\partial t} \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\operatorname{Tr}((W^* + tZ)^T A_{i,j} (W^* + tZ))}{2\sigma^2}} A_{i,j} W^* && (22) \\
= & \lim_{t \to 0} \frac{\partial}{\partial t} \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\operatorname{Tr}((W^{*T} A_{i,j} W^* + tZ^T A_{i,j} W^* + tW^{*T} A_{i,j} Z + t^2 Z^T A_{i,j} Z)}{2\sigma^2}} A_{i,j} W^* && (23) \\
= & -\sum_{i,j} \frac{\gamma_{i,j}}{2\sigma^4} e^{-\frac{\operatorname{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} \operatorname{Tr}(Z^T A_{i,j} W^* + W^{*T} A_{i,j} Z) A_{i,j} W^* && (24) \\
= & -\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^4} e^{-\frac{\operatorname{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} \operatorname{Tr}(Z^T A_{i,j} W^*) A_{i,j} W^* \qquad \text{as } A_{i,j} = A_{i,j}^T, && (25) \\
T_2 = & \lim_{t \to 0} \frac{\partial}{\partial t} \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} t e^{-\frac{\operatorname{Tr}((W^* + tZ)^T A_{i,j} (W^* + tZ))}{2\sigma^2}} A_{i,j} Z && (26) \\
= & \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\operatorname{Tr}(W^{*T} A_{i,j} W^*)}{2\sigma^2}} A_{i,j} Z, && (27) \\
T_3 = & \lim_{t \to 0} \frac{\partial}{\partial t} (W^* + tZ)\Lambda && (28) \\
= & Z\Lambda. && (29)
\end{aligned}$$

Hence, putting all three terms together yields

$$\mathcal{D}\nabla\mathcal{L}[Z] = \left\{ \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}(({W^*}^T A_{i,j} W^*)}{2\sigma^2}} \left[A_{i,j} Z - \frac{1}{\sigma^2} \mathrm{Tr}(Z^T A_{i,j} W^*) A_{i,j} W^* \right] \right\} - Z\Lambda. \quad (30)$$

Hence,

$$\mathrm{Tr}(Z^T \nabla^2_{W^* W^*} \mathcal{L}(W^*, \Lambda^*) Z) = \mathrm{Tr}(Z^T \mathcal{D}\nabla\mathcal{L}[Z]), \quad (31)$$

$$= \left\{ \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}(({W^*}^T A_{i,j} W^*)}{2\sigma^2}} \left[\mathrm{Tr}(Z^T A_{i,j} Z) - \frac{1}{\sigma^2} \mathrm{Tr}(Z^T A_{i,j} W^*)^2 \right] \right\} - \mathrm{Tr}(Z^T Z \Lambda_W). \quad (32)$$

Next, let Z be such that $Z \neq 0$ and $\nabla h(W^*)^T Z = 0$, where

$$h(W^*) = {W^*}^T W^* - I. \quad (33)$$

Therefore, the constraint condition can be written on Z in (11) can be written as

$$\begin{aligned} \nabla h(W^*)^T Z = & \begin{array}{c} \lim \\ t \to 0 \end{array} \frac{\partial}{\partial t} \frac{(W^* + tZ)^T (W^* + tZ) - {W^*}^T W^*}{t} \\ & = Z^T W^* + {W^*}^T Z = 0. \end{aligned} \quad (34)$$

Using Equations (32) and (34) lemma 4.1 follows. □

Recall from Lemma 2 that W^* consists of the q eigenvectors of $\Phi(W^*)$ with the smallest eigenvalues. We define $\bar{W}^* \in \mathbb{R}^{d \times d-q}$ as all other eigenvectors of $\Phi(W^*)$. Because Z has the same dimension as W^*, each column of Z resides in the space of $\mathbb{R}^d$. Since the eigenvectors of $\Phi(W^*)$ span $\mathbb{R}^d$, each column of Z can be represented as a linear combination of the eigenvectors of $\Phi(W^*)$. In other words, each column z_i can therefore be written as $z_i = W^* P_W^{(i)} + \bar{W}^* P_{\bar{W}^*}^{(i)}$, where $P_{W^*}^{(i)} \in \mathbb{R}^{q \times 1}$ and $P_{\bar{W}^*}^{(i)} \in \mathbb{R}^{d-q \times 1}$ represents the coordinates for the two sets of eigenvectors. Using the same notation, we also define $\Lambda^* \in \mathbb{R}^{q \times q}$ as the eigenvalues corresponding to W^* and $\bar{\Lambda^*} \in \mathbb{R}^{d-q \times d-q}$ as the eigenvalues corresponding to $\bar{W}^*$. The entire matrix Z can therefore be represented as

$$Z = \bar{W}^* P_{\bar{W}^*} + W^* P_{W^*}. \quad (35)$$

Furthermore, it can be easily shown that P_{W^*} is a skew symmetric matrix, or $-P_{W^*} = P_{W^*}^T$. By setting Z from Equation (20) into (35), the constraint can be rewritten as

$$[P_{\bar{W}^*}^T {\bar{W}^*}^T + {P_W^*}^T {W^*}^T] W^* + {W^*}^T [\bar{W}^* P_{\bar{W}^*} + W^* P_{W^*}] = 0. \quad (36)$$

Simplifying the equation yields the relationship

$${P_W^*}^T + P_{W^*} = 0. \quad (37)$$

Using these definitions, we define the following sublemma.

Lemma 4.2. *Given a fixed point W^* and a Z satisfying condition (20), the condition $\mathrm{Tr}(Z^T \mathcal{D}\nabla\mathcal{L}[Z]) \geq 0$ is equivalent to*

$$\mathrm{Tr}(P_{\bar{W}^*}^T \bar{\Lambda^*} P_{\bar{W}^*}) - \mathrm{Tr}(P_{\bar{W}^*} \Lambda^* P_{\bar{W}^*}^T) \geq C_2, \quad (38)$$

where

$$C_2 = \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^4} e^{-\frac{\mathrm{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} \mathrm{Tr}(Z^T A_{i,j} W^*)^2, \tag{39}$$

$P_{W^*}, P_{\bar{W}^*}$ are given by Equation (35), and $\Lambda^*, \bar{\Lambda}^*$ are the diagonal matrices containing the bottom and top eigenvalues of $\Phi(W^*)$ respectively.

Proof. By condition (19),

$$\mathrm{Tr}(Z^T \mathcal{D}\nabla\mathcal{L}[Z]) = C_1 - C_2 + C_3, \tag{40}$$

where

$$\begin{aligned}
C_1 &= \mathrm{Tr}\left(Z^T \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} A_{i,j} Z \right), \\
C_2 &= \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^4} e^{-\frac{\mathrm{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} \mathrm{Tr}(Z^T A_{i,j} W^*)^2, \\
C_3 &= -\,\mathrm{Tr}(Z^T Z \Lambda^*).
\end{aligned}$$

C_1 can be written as

$$\begin{aligned}
C_1 &= \mathrm{Tr}\left(Z^T \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}((W^{*T} A_{i,j} W^*)}{2\sigma^2}} A_{i,j} Z \right) \\
&= \mathrm{Tr}(Z^T \Phi(W^*)[\bar{W}^* P_{\bar{W}^*} + W^* P_{W^*}]) \\
&= \mathrm{Tr}(Z^T [\Phi(W^*)\bar{W}^* P_{\bar{W}^*} + \Phi(W^*) W^* P_{W^*}]) \\
&= \mathrm{Tr}(Z^T [\bar{W}^* \bar{\Lambda} P_{\bar{W}^*} + W^* \Lambda P_{W^*}]) && \text{By definition of eigenvalues.} \\
&= \mathrm{Tr}([P_{\bar{W}^*}^T \bar{W}^{*T} + P_W^{*T} W^{*T}][\bar{W}^* \bar{\Lambda} P_{\bar{W}^*} + W^* \Lambda P_{W^*}]) && \text{Substitute for } Z \\
&= \mathrm{Tr}(P_{\bar{W}^*}^T \bar{\Lambda} P_{\bar{W}^*}) + \mathrm{Tr}(P_{W^*}^T \Lambda P_W) && \text{Given } W^{*T} W^* = I, \bar{W}^{*T} W^* = 0.
\end{aligned}$$

Similarly

$$\begin{aligned}
C_3 &= -\,\mathrm{Tr}(Z^T Z \Lambda) \\
&= -\,\mathrm{Tr}([P_{\bar{W}^*}^T \bar{W}^{*T} + P_{W^*}^T W^{*T}][\bar{W}^* P_{\bar{W}^*} + W^* P_{W^*}]\Lambda) \\
&= -\,\mathrm{Tr}([P_{\bar{W}^*}^T P_{\bar{W}^*} + P_{W^*}^T P_{W^*}]\Lambda) \\
&= -\,\mathrm{Tr}(P_{\bar{W}^*}^T P_{\bar{W}^*} \Lambda) - \mathrm{Tr}(P_{W^*}^T P_{W^*} \Lambda).
\end{aligned}$$

Because P_{W^*} is a square skew symmetric matrix, the diagonal elements of $P_{W^*} P_{W^*}^T$ is the same as the diagonal of $P_{W^*} P_{W^*}^T$. From this observation, we conclude that $\mathrm{Tr}(P_{W^*} P_{W^*}^T \Lambda) = \mathrm{Tr}(P_{W^*}^T P_{W^*} \Lambda)$. Hence,

$$C_3 = -\,\mathrm{Tr}(P_{\bar{W}^*} \Lambda P_{\bar{W}^*}^T) - \mathrm{Tr}(P_{W^*}^T \Lambda P_{W^*}).$$

Putting all 3 parts together yields

$$\begin{aligned}
\mathrm{Tr}(Z^T \mathcal{D}\nabla\mathcal{L}[Z]) &= \mathrm{Tr}(P_{\bar{W}^*}^T \bar{\Lambda} P_{\bar{W}^*}) + \mathrm{Tr}(P_{W^*}^T \Lambda P_{W^*}) - C_2 - \mathrm{Tr}(P_{\bar{W}^*} \Lambda P_{\bar{W}^*}^T) - \mathrm{Tr}(P_{W^*}^T \Lambda P_{W^*}) \\
&= \mathrm{Tr}(P_{\bar{W}^*}^T \bar{\Lambda} P_{\bar{W}^*}) - \mathrm{Tr}(P_{\bar{W}^*} \Lambda P_{\bar{W}^*}^T) - C_2.
\end{aligned} \tag{41}$$

The 2nd order condition (11) is, therefore, satisfied, when

$$\operatorname{Tr}(P_{\bar{W}^*}^T \bar{\Lambda} P_{\bar{W}^*}) - \operatorname{Tr}(P_{\bar{W}^*} \Lambda P_{\bar{W}^*}^T) \geq C_2. \tag{42}$$

□

Lemma 4.3. *Given W^*,$\bar{W}^*$,$\bar{\Lambda}^*$, and Λ^* as defined in Equation (35), if the corresponding smallest eigenvalue of $\bar{\Lambda}^*$ is larger than the largest eigenvalue of Λ^*, then given a large enough σ the condition (11) of Lemma 1 is satisfied.*

Proof. To proof sublemma (4.3), we provide bounds on each of the terms in (42). Starting with C_2 defined at (39). It has a trace term, $(\operatorname{Tr}(Z^T A_{ij} W^*))^2$ that can be rewritten as

$$(\operatorname{Tr}(A_{ij} W^* Z^T))^2 = (\operatorname{Tr}(A_{ij} W^* P_{W^*}^T {W^*}^T + A_{ij} W^* P_{\bar{W}^*}^T \bar{W^*}^T))^2. \tag{43}$$

Since A_{ij} is symmetric and $W^* P_{W^*}^T {W^*}^T$ is skew-symmetric, then $\operatorname{Tr}(A_{ij} W^* P_{W^*}^T {W^*}^T) = 0$. Hence

$$\begin{aligned}(\operatorname{Tr}(Z^T A_{ij} W^*))^2 = (\operatorname{Tr}(A_{ij} W^* Z^T))^2 &= (\operatorname{Tr}(A_{ij} W^* P_{\bar{W}^*}^T \bar{W^*}^T))^2 &(44)\\ &\leq \operatorname{Tr}(A_{i,j}^T A_{ij}) \operatorname{Tr}(P_{\bar{W}^*}^T P_{\bar{W}^*}) &(45)\end{aligned}$$

where the last inequality follows from Cauchy-Schwartz inequality and that fact that ${W^*}^T W^* = I$ and $\bar{W^*}^T \bar{W}^* = I$. Thus, C_2 in (41) is bounded by

$$C_2 \leq \sum_{i,j} \frac{|\gamma_{i,j}|}{\sigma^4} e^{-\frac{\operatorname{Tr}(({W^*}^T A_{i,j} W^*)}{2\sigma^2}} \operatorname{Tr}(A_{i,j}^T A_{ij}) \operatorname{Tr}(P_{\bar{W}^*}^T P_{\bar{W}^*}) \tag{46}$$

Similarly, the remaining terms in (40) can be bounded by

$$C_1 = \operatorname{Tr}(P_{\bar{W}^*}^T \bar{\Lambda}^* P_{\bar{W}^*}) \geq \min_i(\bar{\Lambda}^*{}_i) \operatorname{Tr}(P_{\bar{W}^*} P_{\bar{W}^*}^T) \tag{47}$$

$$C_3 = -\operatorname{Tr}(P_{\bar{W}^*} \Lambda^* P_{\bar{W}^*}^T) \geq -\max_i(\Lambda_i^*) \operatorname{Tr}(P_{\bar{W}^*}^T P_{\bar{W}^*}). \tag{48}$$

Using the bounds for each term, the Equation (42) can be rewritten as

$$[\min_i(\bar{\Lambda}^*{}_i) - \max_j(\Lambda_j^*)] \operatorname{Tr}(P_{\bar{W}^*}^T P_{\bar{W}^*}) \geq \sum_{i,j} \frac{|\gamma_{i,j}|}{\sigma^4} e^{-\frac{\operatorname{Tr}(({W^*}^T A_{i,j} W^*)}{2\sigma^2}} \operatorname{Tr}(P_{\bar{W}^*}^T P_{\bar{W}^*}) \tag{49}$$

$$[\min_i(\bar{\Lambda}^*{}_i) - \max_j(\Lambda_j^*)] \geq \sum_{i,j} \frac{|\gamma_{i,j}|}{\sigma^4} e^{-\frac{\operatorname{Tr}(({W^*}^T A_{i,j} W^*)}{2\sigma^2}} \tag{50}$$

It should be noted that Λ^* is a function of $\frac{1}{\sigma^2}$. This relationship could be removed by multiplying both sides of the inequality by σ^* to yield

$$\sigma^2[\min_i(\bar{\Lambda}^*{}_i) - \max_j(\Lambda_j^*)] \geq \sum_{i,j} \frac{|\gamma_{i,j}|}{\sigma^2} e^{-\frac{\operatorname{Tr}(({W^*}^T A_{i,j} W^*)}{2\sigma^2}}. \tag{51}$$

Since σ^2 is always a positive value, as long as all the eigenvalues from $\bar{\Lambda}^*$ is larger than all the eigenvalues from Λ^*, the left hand side of the equation will always be greater than 0. As $\sigma \to \infty$, the right hand side approaches 0, and the condition (11) of Lemma 1 is satisfied.

□

Appendix E Convergence Plot from Experiments

Figure 4 summarizes the convergence activity of various experiments. For each experiment, the top figure provides the magnitude of the objective function. It can be seen that the values converges towards a fixed point. The middle plot provide updates of the gradient of the Lagrangian. It can be seen that the gradient converges towards 0. The bottom plot shows the changes in W during each iteration. The change in W converge towards 0.

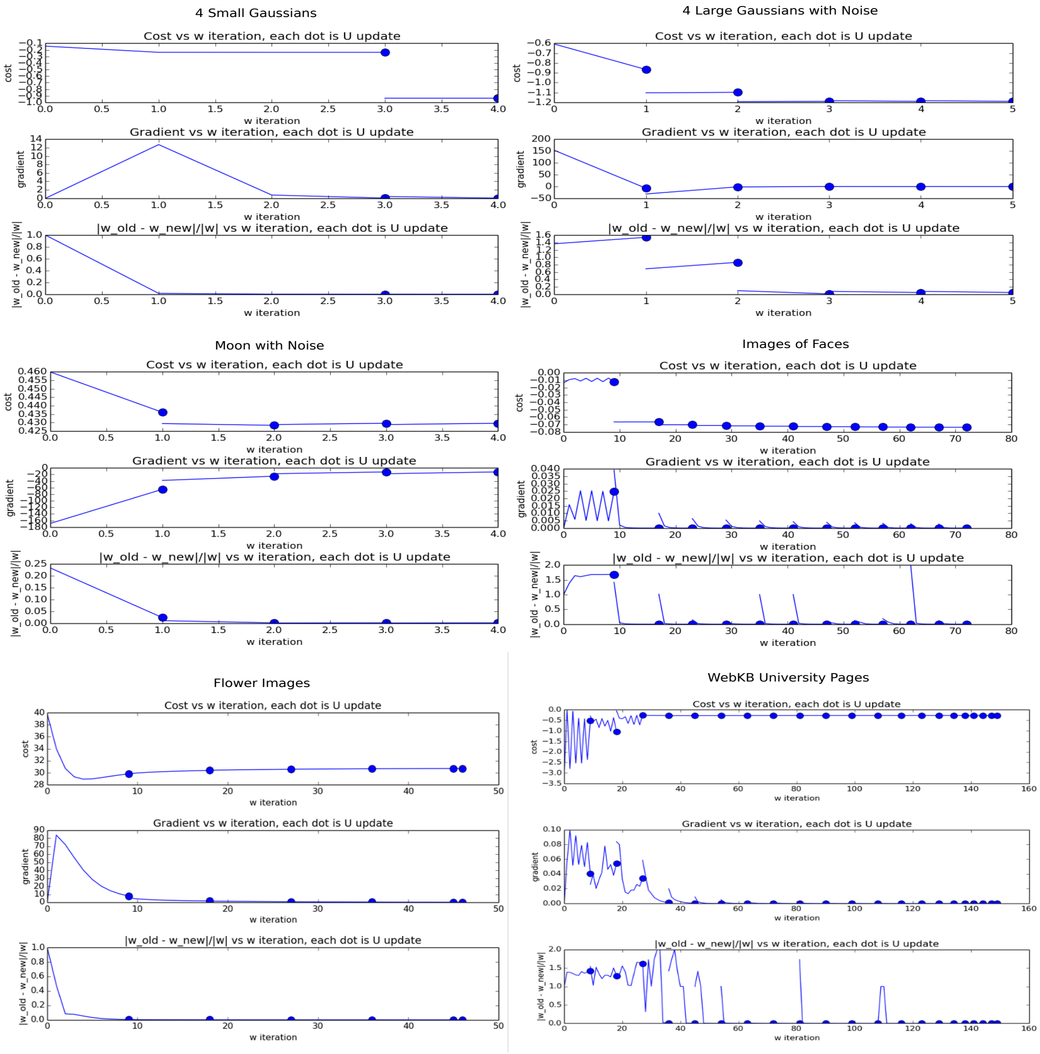

Figure 4: Convergence Results from the Experiments.

Appendix F Proof of Convergence

The convergence property of ISM has been analyzed and yields the following theorem.

Theorem 2. *A sequence $\{W_k\}_{k\in\mathbb{N}}$ generated by Algorithm 2 contains a convergent subsequence.*

Proof. According to Bolzano-Weierstrass theorem, if we can show that the sequences generated from the 1st order relaxation is bounded, it has a convergent subsequence. If we study the Equation $\Phi(W)$ more closely, the key driver of the sequence of W_k is the matrix Φ, therefore, if we can show that if this matrix is bounded, the sequence itself is also bounded. We look inside the construction of the matrix itself.

$$\Phi_{n+1} = \left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}(W_n^T A_{i,j} W_n)}{2\sigma^2}} A_{i,j} \right]$$

From this equation, start with the matrix $A_{i,j} = (x_i - x_j)(x_i - x_j)^T$. Since x_i, x_j are data points that are always centered and scaled to a variance of 1, the size of this matrix is always constrained. It also implies that $A_{i,j}$ is a PSD matrix. From this, the exponential term is always limited between the value of 0 and 1. The value of σ is a constant given from the initialization stage. Lastly, we have the $\gamma_{i,j}$ term. Since $\gamma = D^{-1/2} H (UU^T - \lambda YY^T) H D^{-1/2}$. The degree matrix came from the exponential kernel. Since the kernels are bounded, D is also bounded. The centering matrix H and the previous clustering result Y can be considered as bounded constants. Since the spectral embedding U is a orthonormal matrix, it is always bounded. From this, given that the components of Φ is bounded, the infinity norm of the Φ is always bounded. The eigenvalue matrix of Λ is therefore also bounded. Using the Bolzano-Weierstrass Theorem, the sequence contains a convergent sub-sequence. Given that Φ is a continuous function of W, by continuity, W also has a convergent sub-sequence. □

Appendix G Measure of Non-linear Relationship by HSIC Versus Correlation

The figure below demonstrates a visual comparison of HSIC and correlation. It can be seen that HSIC measures non-linear relationships, while correlation does not.

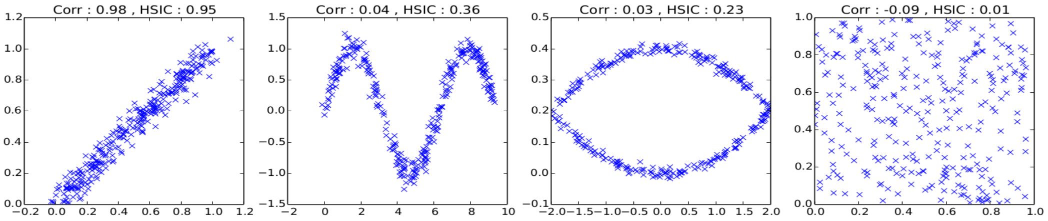

Figure 5: Showing that HSIC captures non-linear information.

Appendix H Implementation Details of the Cost function

The computation of cost function presented in this paper, is a complicated equation that slows down both the implementation and speed of the results.

$$\begin{array}{ll} \underset{W}{\min} & -\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{tr}(W^T A_{i,j} W)}{2\sigma^2}} \\ s.t & W^T W = I \\ & W \in \mathbb{R}^{d \times q} \\ & A \in \mathbb{R}^{d \times d} \\ & \gamma_{i,j} \in \mathbb{R} \end{array} \qquad (52)$$

Instead of solving the function itself, it could be mostly easily done with the following equation.

$$\text{cost} = \text{HSIC}(XW, U) - \lambda\, \text{HSIC}(XW, Y)$$

The simplest way is to write a HSIC function, and pass XW, U, and Y to compute the final cost. Although easy, this approach is not the fastest in terms of separating out the portion of the code that requires constant update, and the portion that remains constant. In this section, a faster approach to implement the cost function is outlined.

Starting with the original cost function :

$$\text{cost} = \text{HSIC}(XW, U) - \lambda\, \text{HSIC}(XW, Y)$$

Convert it into trace format.

$$\text{cost} = \text{Tr}(\tilde{K}HUU^T H) - \lambda\, \text{Tr}(\tilde{K}HYY^T H)$$

Where $\tilde{K}$ is the normalized kernel of XW, which could also be written as $\tilde{K} = D^{-\frac{1}{2}} K_{XW} D^{-\frac{1}{2}}$. Putting this into the cost function.

$$\text{cost} = \text{Tr}\left(D^{-\frac{1}{2}} K_{XW} D^{-\frac{1}{2}} HUU^T H\right) - \lambda\, \text{Tr}\left(D^{-\frac{1}{2}} K_{XW} D^{-\frac{1}{2}} HYY^T H\right)$$

When optimizing U, it is obvious that the 2nd portion does not effect the optimization. Therefore, U can be solved using the following form.

$$U = \underset{U}{\text{argmin}} \quad \text{Tr}(U^T H D^{-1/2} K_{XW} D^{-1/2} HU)$$

The situation get a bit more complicated if we are optimization for W. Using the combination of the rotation property and the combination of the 2 traces, the cost can be written as :

$$\text{cost} = \text{Tr}([D^{-1/2} H(UU^T - \lambda YY^T)HD^{-1/2}]K)$$

In this form, it can be seen that the update of W matrix will only affect the kernel K and the degree matrix D. Therefore, it makes sens to treat the middle portion as a constant which we refer as Ψ.

$$\text{cost} = \text{Tr}([D^{-1/2} \Psi D^{-1/2}]K)$$

Given that $[D^{-1/2} \Psi D^{-1/2}]$ is a symmetric matrix, from this form, we can convert the trace into an element wise product $\odot$.

$$\text{cost} = \sum_{i,j} ([D^{-1/2} \Psi D^{-1/2}] \odot K)_{i,j}$$

To further reduction the amount of operation, we let d be a vector of the diagonal elements of $D^{-1/2}$, hence $d = \text{diag}(D^{-1/2})$, this equality hold.

$$D^{-1/2} \Psi D^{-1/2} = [dd^T] \odot \Psi$$

Therefore, the final cost function can be written in its simplest form as :

$$\text{cost} = \sum_{i,j} \Gamma_{i,j} = \sum_{i,j} (\Psi \odot [dd^T] \odot K)_{i,j}$$

During update, as W update during each iteration, the matrix Ψ stays as a constant while dd^T and K update. The benefit of this form minimize the complexity of the equation, while simplify cost into easily parallelizable matrix multiplications. The equation also clearly separates the elements into portions that require an update and portions that does not.

Appendix I Implementation Details of the Derivative

As it was shown from previous sections, the gradient of our cost function using the Gaussian Kernel has the following form.

$$\nabla f(W) = \left[\frac{1}{\sigma^2} \sum \gamma_{i,j} K_{i,j} A_{i,j} \right] W$$

It is often shown as :

$$\nabla f(W) = \Phi W$$

The key is therefore to find Φ.

$$\Phi = \frac{1}{\sigma^2} \sum \gamma_{i,j} K_{i,j} A_{i,j}$$

If we note that $A_{i,j} = (x_i - x_j)(x_i - x_j)^T$. It can be seen that the inner portion is identical to the cost function. The difference is the addition of the $A_{i,j}$ matrix and a constant of $\frac{1}{\sigma^2}$. These extra factors can be incorporate in the following form.

$$\Phi = \frac{1}{\sigma^2} Q^T \operatorname{diag}(\operatorname{Vec}(\Gamma)) Q$$

Where :

$$Q = (X \otimes \mathbf{1}_n) - (\mathbf{1}_n \otimes X)$$

Note that $\otimes$ is a tensor product and $\mathbf{1}_n$ is a 1 vector with a length of n.

And :

$$\Gamma = \Psi \odot [dd^T] \odot K$$

Since Q is a constant that never changes during the optimization, it could be calculated at the beginning and cached. During each W update using the gradient, Ψ and Q are considered as constants while K and D require a constant update. However, each time the U matrix is updated, Ψ must also be updated.

Here we outline the Algorithm for the W optimization update scheme.

1. Initialize $W_0 = 0$ for the first time, and W_k if U has been updated.

2. Calculate Q, Ψ and store them as constants

3. Calculate K, D, Φ

4. $W_{k+1} = \overrightarrow{\text{eig}}_{\min}(\Phi)$, pick q least dominant eigenvectors as W_{k+1}.

5. Repeat 3,4 until W convergence