Visualization, kernels and subspaces: a practical study

Adriano Oliveira Barbosa
ICMC–USP / FACET–UFGD
Email: adrianobarbosa@ufgd.edu.br

Luis Gustavo Nonato
ICMC–USP
Email: gnonato@icmc.usp.br

***Abstract*—High-dimensional data are typically handled as laying in a single subspace of the original space. However, data involved in real applications are usually spread around in distinct subspaces which may have different dimensions. We would like to study how the subspace structure information can be used to improve visualization tasks. On the other hand, what if the data is tangled in this high-dimensional space, how to visualize it's patterns or how to accomplish classification tasks? One could, for example, map the data in another high-dimensional space using a mapping capable of untangle the data making the patterns clear, rendering the visualization or classification an easy task. This paper presents an study for both problems pointed out above. For the former, we use subspace clustering techniques to define, when it exists, a subspace structure, studying how this information can be used to support visualization tasks based on multidimensional projections. For the latter problem we employ kernel methods, well known in the literature, as a tool to assist visualization tasks. We use a similarity measure given by the kernel to develop a completely new multidimensional projection technique capable of dealing with data embedded in the implicit feature space defined by the kernel.**

I. Introduction

High-dimensional data can be aquired from many sources. This kind of data is usually described using coordinates in a cartesian high-dimensional space ($\mathbb{R}^n$ with big n). As these spaces can't be visualized directly, we use mathematical and computational tools, such as multidimensional projection, wich are able to process and present this kind of data in a compreensive way.

Some multidimensional projection techniques assume the data is embedded completelly (or likelly) in a single subspace with dimension smaller than the number of attributes in the data. However, there is no reason to believe this afirmation is always true. To deal with data laying in multiple independent subspaces we use subspace clustering techniques [1]. The goal of this class of techniques is to segment the data in clusters each one belonging to a different subspace. These techniques can be divided into algebraic [2]–[4], iterative [5]–[7], statistical [8], [9], and spectral methods [10], [11]. Low-Rank Representation (LRR) [10] is a state of the art technique, which aims to find a low-rank representation of the data by solving for a matrix with minimum rank, subject to some restrictions. Some visualization techniques are designed to perform optimally in labeled data. Linear Discrimant Analysis (LDA) [12] computes a linear combination of the data attributes wich allow us to find the subspace with best separability between labels. We use this techniques in our implementation. In Section II we study how effective could be the combination of the subspace information with multidimensional projection in visualization tasks.

Another aproach while studying high-dimensional data can be made using kernel methods [13]. The data is implicitly embedded in a higher-dimensional space (feature space), making patterns and intrinsic structures in the data more clear and easilly segmented. A kernel functions is a similarity measure in this higher-dimensional space. These measures allow us to study the data without knowing the implicit map in the feature space. Most multidimensional projection methods are able to map data from similarity information either are not flexible enough as to user interaction or do not scale properly to large data sets [14]–[17]. Existing interactive and computationally efficient methods can only handle data embedded in a Cartesian space [18], [19], what considerably restricts their applicability. We propose a technique called Kernel-based Linear Projection (Kelp) wich is abble to deal with the data laying in the feature space to fill this gap, since it is computationally efficient, enables interactive manipulation of the projection layouts and it is able to handle kernelized data (data embedded in the feature space). Kelp relies on a solid mathematical formulation, it has low computational cost and enables interactive resources for users dynamically interact with the resulting layout. These desirable properties render Kelp an attractive visualization tool in different scenarios. Moreover, we derive a kernel-based version of differential coordinates which allows for analyzing how neighborhood structures change due to the action of a kernel.

The main contributions of this work[1] are:

- We use subspace clustering and visualization techniques combined for dimensionality reduction;
- A study on the capabilities of subspace segmentation in visualization techniques;
- A modification on the LAMP [19] to make use of the segmentation information;
- A novel kernel-based multidimensional projection technique called Kelp, which relies on a solid mathematical formulation to provide a computational efficient visualization to for analyzing kernelized data;

[1]This work is the result of a PhD Thesis

- The use of Kelp as a visualization tool to assist kernel-based applications such as data classification and image segmentation (in the Thesis [20]);
- The combination of *kernel differential coordinates* (also proposed in this work) with Kelp towards understanding how kernel functions affect neighborhood structures during the embedding process. This novel mechanism is a step forward in enabling visualization resources for users comprehend the behavior of kernels.

This paper is organized as follows: Section II describes the study on the effect of subspace clustering and visualization techniques combination; Section III describes the kernel-based technique Kelp; Section IV makes some conclusions about the work.

II. Subspace clustering and multidimensional projection

Suppose the high-dimensional data is drawn from multiple independent linear subspaces, we would like to visualize it's patterns and intrinsic structures. Using LRR we compute the subspaces and find the data membership. The next step is to use the subspace membership information found by LRR as labels for LDA, which projects data to a visual space based on the labels. LDA performs the projection using a linear mapping that separates data with distinct labels.

A. Subspace Clustering

Given a data set $X = \{x_1, x_2, \ldots, x_m\}$, $x_i \in \mathbb{R}^d$, with m instances in a d-dimensional space, suppose the data can be decomposed as $X = X_0 + E_0$, where X_0 are the data drawn from the independent linear subspaces and E_0 is the "error" of the data, due to corruptions such as noise or outliers. The LRR method aims to find a low-rank matrix X_0 from the given data set X corrupted by errors E_0. The solution is given by the following minimization problem:

$$\min_{Z,E} \text{rank}(Z) + \lambda \|E\|_l, \text{ s.t. } X = AZ + E, \qquad (1)$$

where A is a "basis" for the space where the data lie, and $\|\cdot\|_l$ is a matrix norm that may vary depending on what kind of error we wish to filter.

As the problem (1) may not have a unique solution, the following problem is solved instead:

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_l, \text{ s.t. } X = AZ + E,$$

where $\|\cdot\|_*$ is the nuclear norm (sum of singular values).

When the data are affected by sample-specific corruptions or outliers, we can use the norm $\|E\|_{2,1} = \sum_{j=1}^{m} \sqrt{\sum_{i=1}^{m} \|E_{ij}\|^2}$, which is sensible to this kind of error [10].

The minimizer Z is a matrix with information about the data membership, and the non-zero entries of the matrix E represent data corruption. Let $Z = UDV^\top$ be the singular value decomposition of Z. To perform the segmentation, we compute the affinity matrix W defined by $W_{ij} = (\tilde{U}\tilde{U}^\top)_{ij}$, where $\tilde{U}$ is formed by $UD^{\frac{1}{2}}$ with normalized rows, and then apply the Normalized Cuts [21] clustering algorithm.

B. LDA Projection

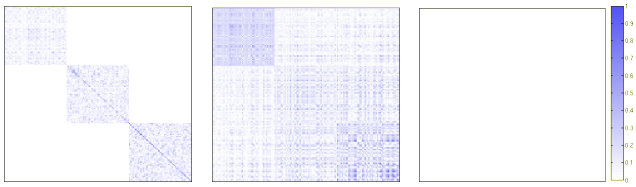
Given data instances $\{x_1, x_2, \ldots, x_m\}$ with labels $\{l_1, l_2, \ldots, l_m\}$, let C_i denote the set of instances with label l_i, m_i the cardinality of C_i, and $\bar{x}_i = \frac{1}{m_i} \sum_{x_j \in C_j} x_j$ be the centroid of C_i. From this, we can compute the "between groups scatter matrix" $S_B = \frac{1}{m} \sum_{j=1}^{m} m_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^\top$ and the "within groups scatter matrix" $S_W = \frac{1}{m} \sum_{l=1}^{k} \sum_{x_j \in C_l} (x_j - \bar{x}_l)(x_j - \bar{x}_l)^\top$, where k is the number of groups, and $\bar{x}$ is the centroid (mean) of the entire data.

Maximization of the objective function $J(P) = \frac{P^\top S_B P}{P^\top S_W P}$ gives us a projection matrix such that the data projection has small variance in each subspace, and large variance between group centroids. This problem is solved by the generalized eigenvalue problem $S_B P = S_W P \Gamma$. Where the columns of P are the eigenvectors associated to the eigenvalues that form the diagonal matrix Γ.

We then use the first p columns of P (which are associated to the p absolutely largest eigenvalues) as the projection matrix to reduce the dimension of the data, mapping them to a p-dimensional space. For visualization, we usually use $p = 2$.

C. Experiments

In our tests use two data sets: an artificially generated consisting of 50 instances of data drawn from $\mathbb{R}^3$, 50 instances drawn from $\mathbb{R}^7$, and 50 instances drawn from $\mathbb{R}^{10}$, and embedded all 150 instances in $\mathbb{R}^{30}$ and the well-known Iris data set [22], wich consists of 150 instances of dimension 4 divided in 3 labels, each one of the species of the flower iris. The results of LRR, with $\lambda = 0.5$, are shown in Figure 1.



(a) Artificial data set Z Matrix (b) Iris data set Z Matrix (c) E matrix in both data sets

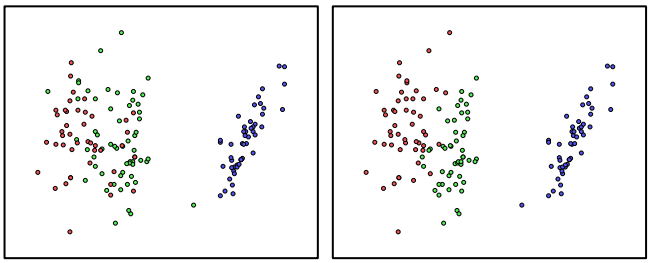
Fig. 1. LRR minimizer on Artificial and Iris data sets.

Notice that for the Artificial data set all subspaces are perfectly identified while for the Iris data set one of the labels is well identified in the block diagonal matrix Z, while the other two labels are more difficult to distinguish. The matrix E is empty in both cases due to the nature of the data sets and because we are using the norm $\|\cdot\|_{2,1}$, which is sensible to sample-specific corruptions.

We projected the Iris data set using LDA with labels given by the LRR segmentation (Figure 2a) and the real data set labels (Figure 2b).

D. Results, Discussion, and Limitations

We evaluate the method by comparing the projections generated by LDA with three other techniques: LAMP [19], t-SNE [23], and a modified version of LAMP. The modification we applied to LAMP is to add the label information for the



(a) Color given by segmentation (b) Real label color

Fig. 2. LDA projection of the Iris data set.

computation of weights. Originally, we had $\alpha_i = \frac{1}{\|x-x_i\|^2}$, where x is the instance to be projected, and x_i is a control point. In the modified version, we have:

$$\alpha_i = \begin{cases} \frac{1}{\|x-x_i\|^2}, & \text{if } x \text{ and } x_i \text{ have the same label} \\ 0, & \text{otherwise.} \end{cases}$$

The quality of the projections generated by our approach is evaluated using four metrics: stress, neighborhood preservation, and two silhouettes. The stress function we use is given by $\frac{1}{\sum_{ij} d_{ij}} \sum_{ij} \left(d_{ij} - \bar{d}_{ij}\right)^2 / d_{ij}^2$, where d_{ij} is the distance in the original space and $\bar{d}_{ij}$ is the distance in the visual space. For each instance of data, the neighborhood preservation measures how many k-nearest neighbors in the original space are among the k-nearest neighbors in the visual space. The silhouette measures cohesion and separation between clusters. It is given by $S = \frac{1}{m} \sum_i \frac{b_i - a_i}{\max\{a_i, b_i\}}$, where a_i (the cohesion) is calculated as the average of the distances between a projected instance y_i (projection of x_i) and all other projected instances belonging to the same cluster as y_i, and b_i (the separation) is the minimum distance between y_i and all other projected instances belonging to other clusters. The silhouette ranges in the interval $[-1, 1]$ and the larger the value of S the better is the cohesion and separation of the data. To compute the silhouette, we need to know the labels of the data. We use both, the real labels (silh1) and the labels given by LRR (silh2) to compute the silhouettes. We use some of the data sets described in the Table II.

Table I summarizes the results. Compared to LAMP, the modified version of LAMP performs better in terms of silh2 (with labels given by LRR), with a small difference in terms of stress. While the stress of LDA is bigger than LAMP and modified LAMP (which is expected, because the objective of LDA is to find the subspace with better separability between labels), it gives a good result in terms of silh2. The results of LDA indicate that the combination of LRR and LDA can be a good choice for dimensionality reduction and unsupervised classification problems where the true label is unknown.

1) Limitations: Subspace clustering techniques assume that the data are drawn from independent subspaces, but this may not be always true in real world data sets. We have extensively tested some examples for many parameters, with no success in finding any reasonable subspace structure. In these cases, we assume that such a subspace structure does not exist and thus that the method cannot be applied properly.

TABLE I

RESULTS, FROM LEFT TO RIGHT THE COLUMNS CORRESPOND TO THE DATA SET NAME, TECHNIQUE AND METRICS: STRESS, NEIGHBORHOOD PRESERVATION, AND SILHOUETTES. BOLD VALUES ARE THE BEST FOR EACH DATA SET AND METRIC.

Data set	Technique	Stress	NP (%)	Silh1	Silh2
Iris	LAMP	**0.0418**	81.8	0.6371	0.3437
	LAMP (M)	0.0791	77.8	0.6032	0.4221
	LDA	0.3095	63.6	0.6889	**0.6758**
	t-SNE	1.71e+6	**86.9**	**0.7633**	0.3392
Synthetic	LAMP	0.0597	80.9	0.8584	0.8584
	LAMP (M)	**0.0521**	82.0	0.9045	0.9045
	LDA	0.0862	85.7	0.9299	0.9299
	t-SNE	6.2266	**89.4**	**0.9956**	**0.9956**
Artificial	LAMP	**0.0539**	85.4	0.6770	0.6770
	LAMP (M)	0.0749	86.0	0.7787	0.7787
	LDA	0.3749	81.2	**0.9492**	**0.9492**
	t-SNE	0.2962	**90.9**	0.8961	0.8961
Wine	LAMP	**0.0383**	90.7	0.2174	0.3629
	LAMP (M)	0.1371	86.9	0.2269	0.4491
	LDA	0.9802	53.3	0.2694	**0.5314**
	t-SNE	0.9312	**94.3**	**0.3139**	0.4262
Mammals	LAMP	**0.0112**	**87.9**	0.9825	0.9825
	LAMP (M)	0.0172	85.5	**0.9924**	**0.9924**
	LDA	1.0000	81.4	0.9311	0.9311
	t-SNE	0.3829	87.7	0.9653	0.9653

III. KERNEL-BASED LINEAR PROJECTION

Our kernel-based multidimensional projection method relies on a subset of samples to perform the mapping. Let $X_s \subset X$, $X_s = \{x_{s_1}, x_{s_2}, \ldots, x_{s_n}\}$ be a subset of samples from X (n accounts for the number of samples while m is the total number of instances in X) and $Y_s = \{y_{s_1}, y_{s_2}, \ldots, y_{s_n}\}$ be the image of X_s in the visual space (Y_s results from the Force Scheme [24] applied to X_s). Lets also denote by K_s the Gram matrix built from X_s, that is, the entries in K_s are given by the values of a kernel function $k : X \times X \to \mathbb{R}$, $k(x_{s_i}, x_{s_j})$.

Suppose that the implicit map $\phi : X \to \mathcal{H}$ associated to the kernel k is known ($\phi(x_{s_i})$ is the embedding of the instance x_{s_i} in the feature space $\mathcal{H}$), our goal is to find a linear mapping $M : \mathcal{H} \to \mathbb{R}^2$ ($\mathbb{R}^2$ being the visual space) such that

$$M\phi(x_{s_i}) = y_{s_i}\,. \tag{2}$$

The linear transformation M should map each sample $\phi(x_{s_i})$ to y_{s_i} in the visual space. The rationale behind the construction above is that, due to linearity, the neighborhood structure of each $\phi(x_{s_i})$ should be preserved by M.

Equation (2) can be written in matrix form as

$$M\Phi = Y. \tag{3}$$

Multiplying both sides of Equation (3) by $\Phi^\top$ we obtain

$$M\Phi\,\Phi^\top = Y\Phi^\top \;\Rightarrow\; nMC_s = Y\Phi^\top\,, \tag{4}$$

where $C_s = \frac{1}{n}\sum_{i=1}^{n} \phi(x_{s_i})\phi(x_{s_i})^\top$ is the covariance matrix computed from the subset of samples X_s. Since C_s is symmetric it can be decomposed as $C_s = UDU^\top$, where the columns of U are the orthonormal eigenvectors $\mathbf{u}_i$ of C_s and D is a diagonal matrix containing the eigenvalues λ_i

as diagonal elements. The pseudo inverse of C_s is given by $C_s^+ = U\widetilde{D}^{-1}U^\top$, being $\widetilde{D}^{-1}$ the inverse of nonzero diagonal elements in D. Applying the pseudo inverse in Equation (4) results in:

$$M = \frac{1}{n} Y \Phi^\top C_s^+ = \frac{1}{n} Y \Phi^\top \left(U \widetilde{D}^{-1} U^\top \right) .$$

The projection of any instance $\phi(x)$ is so given by

$$M\phi(x) = \frac{1}{n} Y \Phi^\top U \widetilde{D}^{-1} U^\top \phi(x) . \tag{5}$$

Let A be the matrix with columns formed by eigenvectors $\mathbf{a}_i$ of K_s and, making an abuse of notation, let U be now the matrix containing only the eigenvectors of C_s associated to nonzero eigenvalues. In the Thesis [20], we prove that

$$U = \Phi A \;\; \Rightarrow \;\; \Phi^\top U = \Phi^\top \Phi A \;\; \Rightarrow \;\; \Phi^\top U = K_s A \tag{6}$$

and

$$U^\top \phi(x) = (\Phi A)^\top \phi(x) = A^\top \Phi^\top \phi(x) = A^\top \mathbf{k}_x , \tag{7}$$

where $\mathbf{k}_x = (k(x, x_{s_1}),\ k(x, x_{s_2}), \ldots,\ k(x, x_{s_n}))^\top$.

Using the fact that the eigenvalues of C_s and K_s relate to each other according to $\gamma_i = n\lambda_i$ (also proved in the Thesis [20]), where γ_i are the eigenvalues of K_s, and using Equations (6) and (7) in Equation (5) we have

$$M\phi(x) = Y K_s A \Gamma^{-1} A^\top \mathbf{k}_x , \tag{8}$$

where Γ^{-1} is the diagonal matrix with elements $1/\gamma_i$.

Notice that the term on the right in Equation (8) involves only known quantities. In fact, Y is the matrix containing the coordinates of the samples in the visual space, K_s is the Gram matrix built from X_s, matrix A has columns given by eigenvectors of K_s, diagonal elements in Γ^{-1} are the inverse of the eigenvalues of K_s, and the vector $\mathbf{k}_x$ is made up of kernel values between x and x_{s_i}, where x is an instance to be projected. Therefore, given the samples, their image in the visual space, and the kernel $k(x, x_{s_i})$, we project any data instance x_i from X to the visual space by simply evaluating Equation (8) in $x = x_i$. In fact, besides Y_s, only $k(\cdot, \cdot)$ need to be known to accomplish the projection of X.

A. Experiments and results

The quality of Kelp is attested through two different sets of comparisons. The first set assesses Kelp's performance as to accuracy and computational time. Kelp is compared against 5 existing techniques employing 8 data sets which vary considerably in terms of size and dimensionality (see Table II). Techniques employed in the comparisons were chosen because they share similarities with Kelp, namely, they also rely on a subset of samples to perform the projection and can deal with kernelized data (data in the feature space). More specifically, Fastmap [25], Hybrid [26], Landmark MDS [27], Pekalska [28], and PLP [29] are methods that present a good performance in terms of stress/time.

The blue box plots in Figure 3 show the range of stress obtained by Kelp and the other techniques when mapping the data sets in Table II. One can easily see that Kelp is one

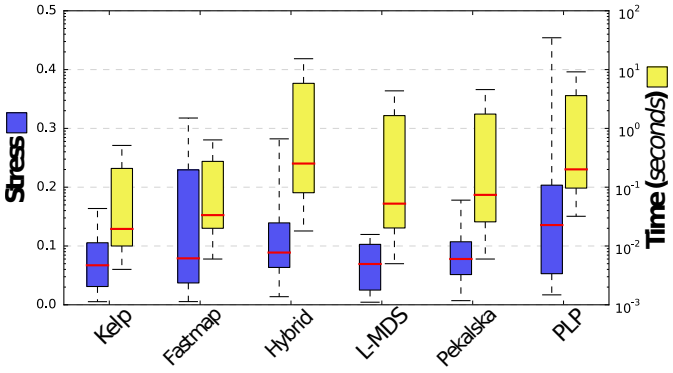
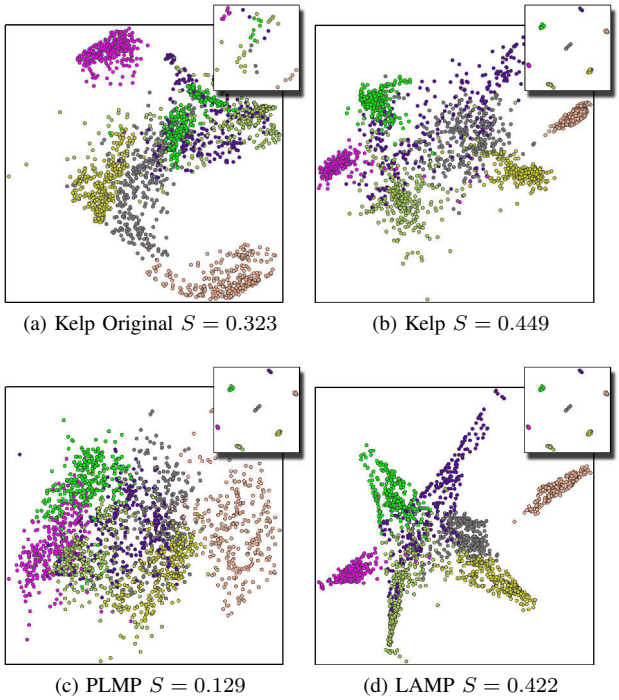


Fig. 3. Box plot of stress and time for data sets in Table II.

of the most accurate technique, being comparable to highly precise methods such as Landmark MDS and Pekalska. Box plots in yellow show that Kelp also performs well in terms of computational times, being comparable to Fastmap, which is well known for its computational efficiency. Notice that Kelp is almost one order of magnitude faster than Landmark MDS and Pekalska, the two methods comparable to Kelp in terms of accuracy.

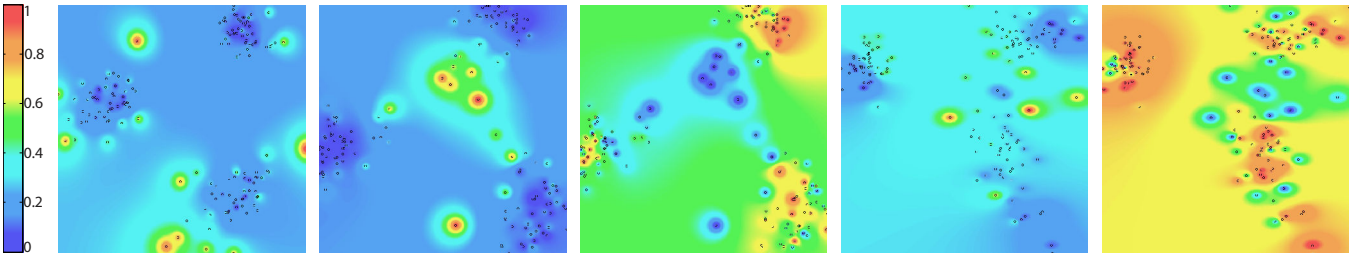


(a) Kelp Original $S = 0.323$ (b) Kelp $S = 0.449$

(c) PLMP $S = 0.129$ (d) LAMP $S = 0.422$

Fig. 4. Comparing Kelp's sensitivity as to user interaction. The upper right insets show the position of the samples.

Kelp's sensitivity with respect to user intervention is analyzed in Figure 4. Figure 4a shows the projection produced by Kelp when samples are arranged in the visual space by the Force Scheme. The top right inset depicts the position of the samples after applying the Force Scheme to a subset of randomly selected samples. Figures 4b, 4c, and 4d show the layouts produced by Kelp, PLMP, and LAMP respectively, after user intervention, that is, user has manually grouped samples accordingly to their groups so as to better define clusters in the visual space (see the top right insets). Notice



(a) $\|\delta_i\|$ (PLMP) (b) $\|\delta_{\phi_i}\|$ (Kelp-Gaussian) (c) $\frac{\|\delta_i\|}{\|\delta_{\phi_i}\|}$ (Kelp-Gaussian) (d) $\|\delta_{\phi_i}\|$ (Kelp-$\langle\cdot,\cdot\rangle^2$) (e) $\frac{\|\delta_i\|}{\|\delta_{\phi_i}\|}$ (Kelp-$\langle\cdot,\cdot\rangle^2$)

Fig. 5. Visualizing how a kernel affects neighborhood structures (artificial data set with 150 instances and 4 dimensions): differential coordinate magnitudes in a layout generated by PLMP with Euclidean distance (a) and Kelp using Gaussian (b) and polynomial kernels (d); Magnitude ratio in the Kelp-Gaussian (c) and Kelp-polynomial layouts (e).

that the layout resulting from Kelp has the highest silhouette value, even superior to LAMP, which is known to be quite sensitive to user intervention. Moreover, PLMP and LAMP require data embedded in a Cartesian space, thus they can not be directly employed in kernelized data.

B. Applications

The visualization tool described in this section is a first attempt in visualizing the behavior of kernels and assisting users in kernel-based applications.

Our approach relies on a metric to compare neighborhood structures defined in the original Cartesian space against their counterpart in the feature space induced by a kernel. The metric is defined as follows: let $\delta_i = x_i - \frac{1}{\#N_i}\sum_{j\in N_i} x_j$ be the differential coordinate of x_i, where N_i accounts for the indexes of the k-nearest neighbors of x_i and $\#N_i$ is the cardinality of N_i. The norm $\|\delta_i\|$ is a measure of how far x_i is from the centroid of its neighbors. Let now $\phi(x_i)$ be the image of x_i in a feature space induced by a kernel k. The norm of the differential coordinate δ_{ϕ_i} of $\phi_i = \phi(x_i)$ in the feature space is given by $\|\delta_{\phi_i}\| = \sqrt{\delta_{\phi_i}^\top \delta_{\phi_i}}$ with

$$\|\delta_{\phi_i}\|^2 = k(x_i,x_i) - \frac{2}{\#N_i}\sum_{j\in N_i} k(x_i,x_j) + \frac{1}{(\#N_i)^2}\sum_{j,s\in N_i} k(x_j,x_s) \quad (9)$$

Equation (9) shows that the norm of differential coordinates in feature space can be obtained from kernel values, making it possible to measure how far each instance $\phi(x_i)$ is from the centroid of its neighbors in the feature space. Notice that we are always defining neighborhoods in the Cartesian space, because our goal is to measure how those neighborhoods are affected by the kernel.

Figures 5a and 5b show color maps corresponding to values of $\|\delta_i\|$ and $\|\delta_{\phi_i}\|$ computed in each data instance (artificial data set generated from [30]) in layouts generated by PLMP and Kelp, respectively. Red regions correspond to large values of $\|\delta_i\|$ and $\|\delta_{\phi_i}\|$ while blue colors represent low values, (green color accounts for intermediate values). Notice that that after applying the kernel groups of instances becomes even better defined.

The ratio $\|\delta_i\|/\|\delta_{\phi_i}\|$ measures changes in neighborhood structures when data is embedded in a feature space by the kernel k. Values close to 1 indicate no changes, values close to 0 indicate that instances get farther from their neighbors in a non-symmetric way, and values greater than 1 means that, after applying the kernel, instances become more centralized with respect to their neighbors. Using a transfer function we can visualize the regions where neighborhoods are more affected by the kernel. Figure 5b tells us that the Gaussian kernel better positions instances in terms of their neighbors within the data groups, that is, within the well defined groups, the Gaussian kernel tend to place instances closer to the centroid of their neighbors. However, Figure 5c clearly shows that, when analyzing the ratio between the norm of differential coordinates, red regions (corresponding to values close to zero or greater than one) show up within well defined groups. Since $\|\delta_{\phi_i}\|$ is small within well defined groups (Figure 5b) and the groups have not spread out due to the kernel action, we conclude that the large values of $\|\delta_i\|/\|\delta_{\phi_i}\|$ are due to a tighter grouping produced by the Gaussian kernel. Therefore, as expected, a Gaussian kernel tends to better define the groups.

The same analysis can be performed with kernels other than Gaussian, as illustrated in Figure 5d and 5e. Figure 5d depicts $\|\delta_{\phi_i}\|$ when a polynomial kernel is used to map data to a feature space. Using visualization tool, though, one can see that the polynomial kernel behaves quite similarly to the Gaussian kernel, avoiding to push "outliers" closer to clusters while tightening instances that lie within clusters.

As one can clearly see, differential coordinates turn out to be quite effective to visualize neighborhood changes induced by kernels. It is worth mentioning that, as far as we know, this is the first time that differential coordinates is used to measure neighborhood structures in the context of kernelized data, thus being another contribution of this work.

IV. Conclusion

This paper presented a study of visualization aided by subspace learning and a novel projection technique, called Kelp, designed to map kernelized data to a visual space.

Subspace clustering techniques have been shown to be a promising way to account for the possible intrinsic subspace structure of data. The use of subspace clustering allows us

TABLE II
DATA SETS USED IN THE COMPARISONS, FROM LEFT TO RIGHT THE COLUMNS CORRESPOND TO THE DATA SET NAME, SIZE, DIMENSION (NUMBER OF ATTRIBUTES), AND SOURCE.

	Name	Size	Dim	Source
Section II	Iris	150	4	[22]
	Synthetic	150	4	[30]
	Artificial	150	30	*
	Wine	178	13	[22]
	Mammals	1000	72	[22]
Section III	wdbc	569	30	[22]
	diabetes	768	8	[22]
	segmentation	2,100	19	[22]
	us-countries	3,028	14	[31]
	wine	4,898	11	[22]
	letter rcn	20,000	16	[22]
	mammals	50,000	72	[22]
	viscontest	200,000	10	[32]

to use LDA to perform dimension reduction and classification tasks with good quality in terms of the metrics we have tested. On the other hand, the potential use of Kelp to support kernel-based applications with visualization resources opens new possibilities which could not be efficiently addressed until now. Therefore, flexibility, effectiveness, and ease of implementation render Kelp one of the most attractive multidimensional projection methods for handling kernelized data.

ACKNOWLEDGMENT

The authors acknowledge financial support from CAPES.

PUBLICATIONS AND AWARDS

1) *An initial study on high-dimensional data visualization through subspace clustering.* Sociedade Brasileira de Computação, 2015. (**Honorable Mention Award** from the SIBGRAPI 2015 WVIS) [33]
2) *Visualizing and interacting with kernelized data.* IEEE Transactions on Visualization & Computer Graphics, v. 22, n. 3, p. 1314–1325, 2016. [34]

REFERENCES

[1] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2010.
[2] T. E. Boult and L. G. Brown, "Factorization-based segmentation of motions," in *Proceedings of the IEEE Workshop on Visual Motion*, 1991, pp. 179–186.
[3] J. Costeira and T. Kanade, "A multibody factorization method for independetly moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.
[4] C. W. Gear, "Multibody grouping from motion images," *International Journal of Computer Vision*, vol. 29, no. 2, pp. 133–150, Aug. 1998.
[5] P. K. Agarwal and N. H. Mustafa, "K-means projective clustering," in *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '04. New York, NY, USA: ACM, 2004, pp. 155–165.
[6] P. S. Bradley, O. L. Mangasarian, and P. Pardalos, "k-plane clustering," *Journal of Global Optimization*, vol. 16, no. 1, pp. 249–252, 2000.
[7] P. Tseng, "Nearest q-flat to m points," *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 249–252, 2000.
[8] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.
[9] H. Derksen, Y. Ma, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy coding and compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 1546–1562, 2007.
[10] H. Zhang, Z. Lin, C. Zhang, and J. Gao, "Robust latent low rank representation for subspace clustering," *Neurocomputing*, vol. 145, pp. 369–373, 2014.
[11] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.
[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
[13] B. SchÖlkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
[14] W. S. Torgeson, "Multidimensional scaling of similarity," *Psychometrika*, vol. 30, pp. 379–393, 1965.
[15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[16] H. Hosobe, "A high-dimensional approach to interactive graph visualization," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004, pp. 1253–1257.
[17] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh, "Multigrid multidimensional scaling," *Numerical Linear Algebra with Applications*, vol. 13, pp. 149–171, 2006.
[18] F. V. Paulovich, C. T. Silva, and L. G. Nonato, "Two-phase mapping for projecting massive data sets," *IEEE Trans. on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1281–1290, 2010.
[19] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato, "Local affine multidimensional projection," *IEEE Trans. on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2563–2571, 2011.
[20] A. O. Barbosa, "Visualização, kernels e subespaços: um estudo prático," Ph.D. dissertation, Universidade de São Paulo, 2016.
[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 888–905, 2000.
[22] A. Frank and A. Asuncion. (2010) Uci machine learning repository. [Online]. Available: http://archive.ics.uci.edu/ml
[23] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
[24] E. Tejada, R. Minghim, and L. G. Nonato, "On improved projection techniques to support visual exploration of multidimensional data sets," *Information Visualization*, vol. 2, no. 4, pp. 218–231, 2003.
[25] C. Faloutsos and K. Lin, "FastMap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases," in *ACM SIGMOD*, 1995, pp. 163–174.
[26] F. Jourdan and G. Melancon, "Multiscale hybrid mds," in *Information Visualization*, 2004.
[27] V. de Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Stanford, Tech. Rep., 2004.
[28] E. Pekalska, D. de Ridder, R. P. W. Duin, and M. A. Kraaijveld, "A new method of generalizing Sammon mapping with application to algorithm speed-up," in *Annual Conf. Advanced School for Comput. Imag.*, 1999, pp. 221–228.
[29] F. V. Paulovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato, "Piecewise Laplacian-based projection for interactive data exploration and organization," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1091–1100, 2011.
[30] I. Guyon, "Design of experiments of the NIPS 2003 variable selection benchmark," in *NIPS 2003 workshop on feature extraction and feature selection*, 2003. [Online]. Available: http://www.nipsfsc.ecs.soton.ac.uk/datasets/
[31] B. Shneiderman and J. Seo, "Hierarchical clustering explorer for interactive exploration of multidimensional data," 2008. [Online]. Available: http://www.cs.umd.edu/hcil/hce/examples/application_examples.html
[32] D. Whalen and M. L. Norman, "Competition data set and description," in *2008 IEEE Visualization Design Contest*, 2008. [Online]. Available: http://vis.computer.org/VisWeek2008/vis/contests.html
[33] A. Barbosa, F. Sadlo, and L. G. Nonato, "An initial study on high-dimensional data visualization through subspace clustering." Sociedade Brasileira de Computao, 2015.
[34] A. Barbosa, F. V. Paulovich, A. Paiva, S. Goldenstein, F. Petronetto, and L. G. Nonato, "Visualizing and interacting with kernelized data," *IEEE Transactions on Visualization & Computer Graphics*, vol. 22, no. 3, pp. 1314–1325, 2016.