DOI: 10.1111/cgf.12116
Eurographics Conference on Visualization (EuroVis) 2013
B. Preim, P. Rheingans, and H. Theisel
(Guest Editors)

Volume 32 (2013), Number 3

User-driven Feature Space Transformation

G. M. H. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich

ICMC/USP, São Carlos/SP, Brazil

Abstract

Interactive visualization systems for exploring and manipulating high-dimensional feature spaces have experienced a substantial progress in the last few years. State-of-art methods rely on solid mathematical and computational foundations that enable sophisticated and flexible interactive tools. Current methods are even capable of modifying data attributes during interaction, highlighting regions of potential interest in the feature space, and building visualizations that bring out the relevance of attributes. However, those methodologies rely on complex and non-intuitive interfaces that hamper the free handling of the feature spaces. Moreover, visualizing how neighborhood structures are affected during the space manipulation is also an issue for existing methods. This paper presents a novel visualization-assisted methodology for interacting and transforming data attributes embedded in feature spaces. The proposed approach relies on a combination of multidimensional projections and local transformations to provide an interactive mechanism for modifying attributes. Besides enabling a simple and intuitive visual layout, our approach allows the user to easily observe the changes in neighborhood structures during interaction. The usefulness of our methodology is shown in an application geared to image retrieval.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques— H.5.0 [Information Interfaces and Presentation]: General—

1. Introduction

Most algorithms and computational tools require as input a feature-based representation of the data under processing. Typical examples are computer vision and textual analysis techniques that operate almost exclusively on feature spaces in order to perform tasks such as matching and similarity detection. Feature-spaces are usually generated by applying feature extraction mechanisms to raw data so as to identify attributes that characterize data instances. Attributes are extracted using different approaches and then merged to form the so-called feature-vector, which is interpreted as a point in a high-dimensional space (the feature space). The mapping to feature spaces enables algebraic and geometric operations, thus opening a multitude of possibilities to handle the data.

Feature extraction methods, however, are prone to generate irrelevant attributes that increase the computational burden and affect the accuracy of techniques operating on the data. Automatic and supervised feature (or attribute) selection methods are the remedy commonly used to filter out irrelevant attributes. As the name suggests, automatic feature selection methods avoid user intervention altogether, accomplishing the analysis of attributes based primarily on statistical measures. Supervised schemes rely on training sets to perform the attribute selection, delegating to the user the task of choosing good representatives for the training process. Visualization-assisted interactive methods have become an ally to automatic and supervised attribute manipulation feature selection techniques. Although helpful, visualization-based methods share the common drawback of not providing intuitive mechanisms for free manipulation of data attributes. Additionally they do not depict how neighborhood structures change during attribute manipulation, restricting users' experience throughout the interactive process.

This work presents a novel visualization-assisted interactive methodology for tackling the issues pointed out above, that is, to provide simple and intuitive mechanisms for attribute transformation while allowing users to verify how neighborhood structures change during interaction. The proposed technique relies on a combination of multidimensional projections and orthogonal linear mappings to enable interactive feature space manipulation. More specifically, changes made by the user in the projection layout are mapped back to the feature space so as to modify the dis-

tance relationship amongst a subset of instances. Modified distances are then used to construct local linear mappings that transform the feature space according to user's guidance. Each local mapping is defined as an affine transformation obtained as the solution of an orthogonal minimization problem formulated in terms of user-driven data.

In order to modify the feature space according to user intervention we propose a transfer function that maps the distance between control points (which can be manually positioned by the user) in the visual space to distances in the feature space. Data instances corresponding to control points in the feature space are then rearranged so as to cope with the new distances. The remaining instances are displaced in the feature space according to the position of their closest control points. A local orthogonal affine transformation is built to perform such displacement. The orthogonal mappings are mathematically formulated taking as basis the multidimensional projection technique called LAMP [JCC*11], adapted to map instances from and to the same feature space.

The proposed interaction tool differs from existing visualization-based attribute manipulation approaches in two main aspects. First, it derives from a multidimensional projection mechanism that provides visualization of neighborhood structures during user manipulation. Visualizing neighborhoods of data points makes the interactive process more intuitive, as users can easily figure out which instances are being affected by the space transformation. The feature space transformation combined with the simple but intuitive visualization layout provided by the multidimensional projection render the proposed methodology unique and quite useful for several applications. The usefulness of our approach is shown in an image retrieval interactive attribute manipulation application. The tests we performed for data classification and clustering show that the proposed user-driven feature space transformation method improves the accuracy of classification and clustering algorithms considerably, even in just a few interaction cycles.

In summary, the main contributions of this work are:

- A novel mathematical and computational approach for transforming feature spaces. This new approach relies on force-based scheme and local affine transformations both operating in a high-dimensional feature space;
- An interactive visualization-assisted tool for data attribute manipulation. Besides being intuitive and simple to use, the proposed methodology allows for visualizing how neighborhood structures change during interaction.

To the best of our knowledge this is the first time that multidimensional projection is exploited as a mechanism for transforming high-dimensional data and feature spaces, characterizing another innovative aspect of this work.

2. Related Work

The literature on analysis and selection of attributes from feature-vectors is comprehensive and existing methods vary greatly as to mathematical and computational foundations (see [MBN02] for a survey). In order to contextualize the technique proposed in this work we focus our discussion on techniques that use interactive visualization to assist the analysis and manipulation of feature space.

The simplest mechanism to visualize and interact with feature-vectors is the parallel coordinates plot (PCP) [ID90] and its variants [HW09, Kan00]. Although those techniques have shown useful in cluster discovery and multi-factor analysis, the clutter hampers their use as an intuitive attribute manipulation tool.

More sophisticated interactive methodologies that combine several visualization resources in a single framework have also been proposed. The VaR (Value and Relation) technique by Yang et al. [YHW*07], for example, is an interactive visualization system that combines glyphs, PCPs, pixel-oriented visualization, and multidimensional scaling in a multiple coordinated view enabling interactive resources that support detection of patterns of interest and dependence relation of attributes. The HCE (Hierarchical Clustering Explore) visualization system proposed by Seo and Shneiderman [SS05] also makes use of multiple coordinated views to analyze the correlation among attributes. HCE implements a rank-by-feature mechanism that allows for building plots from user-defined ranking criteria. VHDR [YWRH03] (Visual Hierarchical Dimension Reduction) provides a hierarchical organization of attributes which is visualized in a multiple view window from which one can identify clusters and handle particular data instances. The user assisted technique by Johansson and Johansson [JJ09] provides changing the metric used to evaluate the quality of attributes. Recently Turkay et al. [TFH11] proposed an interactive visualization tool to assist the estimation of intrinsic dimension of data. Their methodology relies on analyzing how the attributes of user selected instances deviate from the global average of those attributes. Although useful for specific applications, techniques described above do not enable fully interactive mechanisms to tweak data attributes towards better characterizing clusters and relevant features.

Multidimensional projections (MP) have emerged as a tool for high-dimensional feature space exploration which is endowed with the intrinsic capability of visually conveying the neighborhood structure of data. Recent approaches such as LAMP [JCC*11] and PLP [PEP*11] further increase such capability by enabling fully interactive data manipulation mechanisms. Despite their flexibility, existing MP methods have not addressed the issue of interactively modifying data attributes. Enabling this functionality is the main goal of the proposed work.

Techniques that provide mechanism for transforming feature spaces have also been described in the literature.

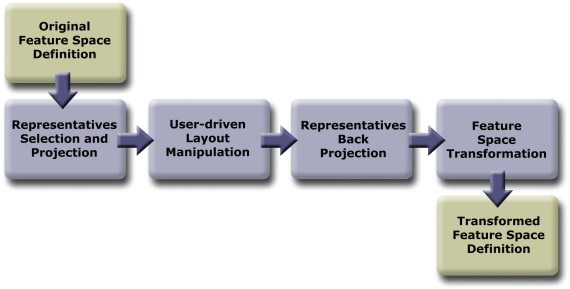

Figure 1: *The proposed method comprises four main steps: representative samples selection and projection, user-driven layout manipulation, back projection of representative samples, and feature space deformation.*

iPCA [JZF*09], for example, provides an interactive interface where the user can manipulate the weights of data attributes in order to figure out which attributes contribute most to the principal component directions. Guo et al. [GWR09] presented a multiple view system that allows the user to examine linear trends in the data as well as to modify attributes so as to impose linear trends. The methodology proposed by Berger et al. [BPFG11] provides powerful interactive resources for local and global attribute modification. Moreover, Berger's methodology enables a continuous navigation mechanism in high-dimensional space. Although attribute manipulation and modification is feasible with methods described above, they rely on complex layouts and functionalities, which can demand some training to make the user familiar with the system. Additionally, those methods do not provide straightforward tools for visualizing how attribute manipulation affects neighborhood structures. Our methodology, in contrast, can easily be operated, since it relies on a simple and intuitive visualization paradigm that intrinsically reveals changes in neighborhood structures and clusters during interaction.

3. Feature Space Manipulation

As illustrated in Figure 1, the proposed methodology begins by selecting representative samples from a dataset (in our experiments these samples are randomly selected). Those representative samples are then projected into the visual space through a distance preserving technique that enables to visualize the neighborhood relation of the samples. The user can manipulate the provided layout by changing the position of representative samples, modifying thus the neighborhood structures in the visual space. After user manipulation, neighborhood structures in the visual space and in the feature space are not in agreement anymore. In order to restore the concordance between them, the set of samples are displaced in the feature space so as to minimize the difference between distances in both spaces. The final transformation of the feature space is performed by a family of local affine mappings built from the new position of representative samples.

The displacement of representative samples and the construction of local affine transformations are the two critical steps of the pipeline illustrated in Figure 1. Mathematical and computational details of those two steps are described below.

3.1. Force Scheme in Feature Space

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be feature-vectors of an m-dimensional dataset containing n instances and $\mathcal{X}_s \subset \mathcal{X}, \mathcal{X}_s = \{x_{s_1}, \ldots, x_{s_k}\}$ be a subset of representative samples. In order to provide a visual representation of the samples, the subset $\mathcal{X}_s$ is mapped to the visual space so that distances are preserved as much as possible. In more mathematical terms, the image y_{s_i} in the visual space of a sample x_{s_i} is computed so as to minimize $|d_m(x_{s_i}, x_{s_j}) - d_2(y_{s_i}, y_{s_j})|$, where $d_m(\cdot,\cdot)$ and $d_2(\cdot,\cdot)$ are distance functions (dissimilarity functions) in the m-dimensional and 2-dimensional spaces, respectively. In this work we use the *Force Scheme* [TMN03] method to compute y_{s_i}, but any accurate and computationally efficient technique could be used.

The set $\mathcal{Y}_s$ ($\mathcal{Y}_s$ is the image of $\mathcal{X}_s$ in the visual space) is the initial layout in the visual space. During interaction, samples $y_{s_i} \in \mathcal{Y}_s$ are moved to new positions $\tilde{y}_{s_i} \in \tilde{\mathcal{Y}}_s$ in the visual space. Observe that we are employing the same notation to represent the image of an instance and its position in space and we will make the distinction clear when necessary. Users can freely modify neighborhood structures during interaction, that is, instances lying far apart may become neighbors after interaction and vice versa.

One of the main goals of our approach is to keep neighborhood structures in unison, ensuring that neighbor instances in the visual space are also neighbors in the high-dimensional space. Therefore, the neighborhood relation created by the user in the visual space must be mapped back to the feature space, displacing representative samples x_{s_i} to new positions $\tilde{x}_{s_i}$.

The main challenge is to redefine distances in feature space to reflect user manipulation. To keep the notation clean, lets denote by d_m^{ij} the distance between instances x_{s_i} and x_{s_j} (feature space) and by d_2^{ij} the distance between y_{s_i} and y_{s_j} (visual space). Denoting by $\tilde{d}_2^{ij}$ the distance between $\tilde{y}_{s_i}$ and $\tilde{y}_{s_j}$ (after user interaction) and assuming that d_2^{ij} and $\tilde{d}_2^{ij}$ have been scaled to range in the same interval as d_m^{ij}, distances in feature space are modified according to the following equation:

$$\tilde{d}_m^{ij} = d_m^{ij}(1 + (\tilde{d}_2^{ij} - d_2^{ij})) \quad (1)$$

where $\tilde{d}_m^{ij}$ is the modified distance.The new position $\tilde{x}_{s_i}$ of

each sample in the feature space is recovered from distances $\tilde{d}_m^{ij}$ by also applying the Force Scheme [TMN03] technique adapted to operate in the m-dimensional feature space.

3.2. Space Transformation

Given the position of the samples $\tilde{x}_{s_i}$ we build upon the idea of local transformations [JCC*11] to compute the new position of the remaining instance $x_i \in \mathcal{X}$ in the (transformed) feature space. More precisely, a local affine transformation $T_{x_i} : \mathbb{R}^m \to \mathbb{R}^m$ is defined to map each x_i, that is, $\tilde{x}_i = T_{x_i}(x_i)$.

The affine transformation $T_{x_i}(x) = xM + t$ associated to x_i is defined so as to minimize:

$$\sum_j \alpha_j \| T_{x_i}(x_{s_j}) - \tilde{x}_{s_j} \|^2, \quad \text{subject to} \quad MM^\top = I, \tag{2}$$

where $\alpha_j = \frac{1}{\|x_{s_j} - x_i\|^2}$, $M^\top$ is the transpose of M, and the $m \times m$ matrix M and vector t are the unknowns. The orthogonality constraint $MM^\top = I$ is imposed to avoid scale and shearing effects. Moreover, the orthogonality constraint guarantees that the eigenvalues of M are equal one (in absolute value), an essential property to ensure stability during interaction, as we further explain in next section.

Some algebraic manipulation allows to write t in terms of M, making possible to express Equation (2) in the matrix form:

$$\| D(AM - B) \|_F, \quad MM^\top = I \tag{3}$$

where $\| \cdot \|_F$ denotes the Frobenius norm, D is a diagonal matrix with entries $D_{ii} = \sqrt{\alpha_i}$, and A and B are $s_k \times m$ matrices with the i-th row given by the vectors $x_{s_i} - \frac{\sum_j \alpha_j x_{s_j}}{\sum_j \alpha_j}$ and $\tilde{x}_{s_i} - \frac{\sum_j \alpha_j \tilde{x}_{s_j}}{\sum_j \alpha_j}$, respectively.

The minimizer of Equation (3) is obtained from:

$$M = UV, \qquad A^\top D^2 B = USV^\top \tag{4}$$

where $D^2 = DD$ and $USV^\top$ is the singular value decomposition of $A^\top D^2 B$.

The transformed position $\tilde{x}_i$ of x_i is given by

$$\tilde{x}_i = \left(x_i - \frac{\sum_j \alpha_j x_{s_j}}{\sum_j \alpha_j} \right) M + \frac{\sum_j \alpha_j \tilde{x}_{s_j}}{\sum_j \alpha_j} \tag{5}$$

An important aspect in the formulation above is that the summation $\sum_j$ does not need to go through all representative samples. In fact, considering only representative samples in a neighborhood of x_i renders the transformation T_{x_i} even more local. Although the overall mapping may become discontinuous when the summation does not traverse the whole set $\mathcal{X}_s$, such discontinuities can help to preserve clusters as well as to concentrate changes in specific regions of the space.

3.3. Interactive Process

As discussed above, the feature space transformation is driven by user manipulation of sample points in the visual space. Interaction is typically initiated with a small set of samples to avoid visual clutter and reduce user effort. However, important structures and clusters may not be properly captured when using a few samples. If the manipulation of the initial set of samples does not result in the expected outcome, an user can successively add new samples to interact with.

In order to alleviate the computational burden and preserve user mental model of the layout the new samples are added without restarting the processing. More precisely, previous samples are kept in place while the new ones are moved by the force-direct scheme. Once the new samples are positioned in the feature space the local mappings described above are used to place the remaining instances.

A new transformation cycle is performed each time new samples are added to the interaction. One important aspect to be observed in the interactive feature space transformation is that the orthogonality constraint imposed in Equation (2) (and Equation (3)) guarantees stability in each cycle of interaction, that is, instances do not converge to a degenerate configuration due to shrinkage or expansion effects. In more mathematical terms, the orthogonality constraint guarantees that the eigenvalues of M are equal one or minus one, and that in turn prevents the instances to collapse or move away after successive cycles of transformations.

4. Results and Applications

In order to show the effectiveness of our feature space manipulation approach we perform transformations in 7 distinct datasets, which vary in size, number of classes and dimensionality, as depicted in Table 1. The *spam* dataset comprises e-mails classified into spam and non-spam. *wdbc* is a breast cancer dataset obtained from digitized images of breast masses. Its instances are classified into two distinct groups, the malignant and benign tumors. The *segmentation* dataset is composed of instances randomly drawn from a database of outdoor images, which were hand-segmented to create a classification for every pixel, resulting in 7 different classes. *shuttle* is composed by log information instances split into 7 different classes. These datasets were retrieved from the *UCI Machine Learning Repository* [FA10]. *caltech* is composed by features extracted using Gabor filters from a selection of 2,937 images from the caltech image collection [FPZ03]. *Imageclef* is composed by features extracted using the the run-length method [LLL88] from 11,744 images of x-rays of different body parts (`http://www.imageclef.org/2009/medical/`). Finally, the *msrcorid* is composed by features obtained by applying Gabor filters to images in a collection of 2,449 photos belonging to 6 different classes [Mic12]. Table 1 summarizes

Table 1: *Summary of the datasets used in our experiments.*

Dataset	size	dim.	number classes
spam	4,601	57	2
wdbc	569	30	2
segmentation	2,100	18	7
shuttle	43,500	9	7
caltech	2,937	48	4
imageclef	11,744	44	5
msrcorid	2,449	48	6

Table 2: *Silhouette of the datasets after one, two and three interaction cycles.*

Dataset	original	$\sqrt{n}/2$	$\sqrt{n}$	$3\sqrt{n}/2$
spam	0.0445	0.0504	0.1374	0.2328
wdbc	0.3412	0.4219	0.5131	0.5919
segmentation	0.2410	0.1763	0.3107	0.3036
shuttle	0.2879	0.5127	0.5775	0.6156
caltech	0.1190	0.2762	0.3364	0.3812
imageclef	0.0305	0.0960	0.1231	0.1228
msrcorid	0.1020	0.1750	0.2668	0.3361

the datasets, showing number of instances, dimensions and classes for each dataset.

Our first experiment shows that the silhouette coefficient [Rou87] of each dataset improves considerably after a few iterations. The silhouette measures both the cohesion and separation between classes. The cohesion a_x of an instance x is calculated as the average distance between x and all other instances in the same class as x. The separation b_x is the minimum distance between x and instances in other classes . The silhouette of a dataset is given by

$$\frac{1}{n}\sum_{x\in\mathcal{X}}\frac{(b_x-a_x)}{max(a_x,b_x)} \tag{6}$$

The silhouette ranges in the interval $[-1,1]$. The larger the value the better is the cohesion and separation of classes. Table 2, shows the original silhouette (second column) of each dataset used in our experiments. We use those values of silhouette as a basis for quantifying the effectiveness of our approach in improving cohesion and separation.

Starting with $\sqrt{n}/2$ samples, which correspond, on average, to less than 1% of the number of instances in the datasets, the user manipulates the samples in the visual space so as to visually group instances belonging to the same class. Notice from the third column in Table 2 that silhouette improves in most cases after the first interaction cycle, that is, after the user to rearrange the initial samples in the visual space. In fact, expressive improvements can be seen in the *shuttle* and *caltech* datasets. Third and fourth columns of Table 2 show silhouette values after the second and third interaction cycles (see Section 3.3), where $\sqrt{n}/2$ new samples were added in each cycle. The silhouette improves 114% for the *shuttle* dataset and more than 220% for the *caltech* and *imageclef* datasets after the third interaction cycle, making clear the effectiveness of our approach to improve the cohesion and separation of instances. Figure 2 shows the resulting silhouettes after 10 iterations, where again $\sqrt{n}/2$ new samples were added in each cycle (iteration 0 indicates the original dataset). Notice that after seven cycles of interaction the silhouette values stabilize, showing that is not necessary to perform many cycles of interaction to improve arrangement of the data in the feature space in terms of group formation and separation.

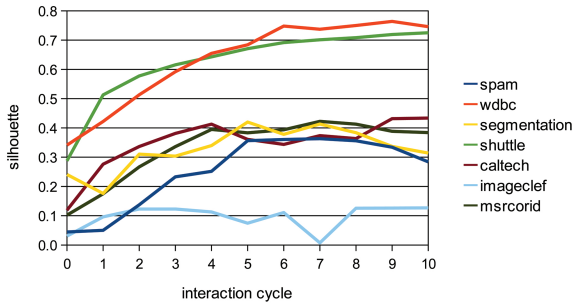

Figure 2: *Silhouette values increase considerably after 10 interaction cycles.*

Classification is an important task for many applications. Typically, classification techniques such as Support Vector Machine (SVM) [CST00] demand a large number of training data in order to work properly. Moreover, such techniques are highly depend on the quality of data attributes. Our second experiment aims at showing that the proposed visualization assisted feature space transformation methodology provides a good alternative to drastically reduce the number of training data as well as lessen the dependency of attributes. Table 3 shows the accuracy [FHOM09] (the degree of correct predictions) of classifications performed with SVM using a radial basis function as kernel – these experiments were executed using the R environment [R C12] which provides an interface to the *libsvm* library [CL11]. Second column in Table 3 brings the accuracy of classifications performed in the original data using $\sqrt{n}/2$ training samples. Columns three to five show the resulting accuracy after one, two and three interaction cycles. Accuracy computation does not take the training sets into account. Notice that a better classification is obtained in all cases after the interactive space transformation. Expressive results have been reached for *caltech* and *msrcorid* datasets. On average, accuracy increases 8% just after the first cycle of interaction, 15% after the second and 23% after the third cycle. The number of training samples used in the third cycle corresponds, on average, to 5%

Table 3: *Accuracy of SVM classification on the original dataset and after interaction cycles.*

Dataset	original	$\sqrt{n}/2$	$\sqrt{n}$	$3\sqrt{n}/2$
spam	0.6060	0.6064	0.6074	0.6457
wdbc	0.6266	0.6266	0.8275	0.9268
segmentation	0.4008	0.3990	0.4247	0.6863
shuttle	0.7842	0.7748	0.7693	0.7586
caltech	0.2540	0.7392	0.7516	0.8169
imageclef	0.4109	0.4490	0.4845	0.5379
msrcorid	0.3800	0.4328	0.6061	0.6790

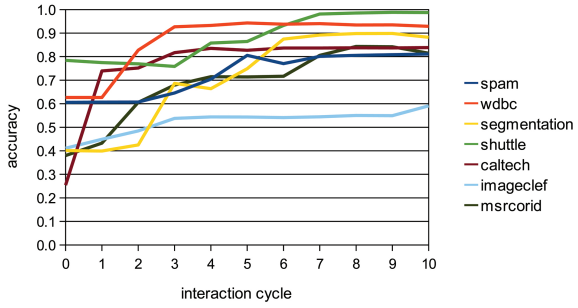

Figure 3: *Accuracy of SVM classification increases as the sample size to create the transformation increases.*

of the instances in the dataset, a number far below that required in typical SVM applications. Figure 3 presents the improvement in terms of accuracy over 10 cycles of interaction. After 10 cycles the accuracy increases 35% on average, achieving an accuracy greater than 0.8 in most of the cases, confirming the effectiveness and usefulness of our approach.

In the tests described above, the accuracy was computed using the existing classification of the data. In a real scenario, where this information does not exist, the initial groups of instances can be obtained by arranging the layout and defining the classes of instances interactively in the visual space. This is an interesting property of our approach, since the visual representation given by the projections can be used to guide the entire process of classification.

We conclude the validation of our approach showing its performance in improving clusters. The analysis is performed using the well known k-means algorithm [Mac67] to create clusters, using the rand index [HA85] to calculate the quality of the results. The R environment is employed and the given class information is used as ground-truth to validate the results. The rand index penalizes both false positive and false negative elements in the clusters, which in mathematical terms can be stated as:

$$RI = \frac{TP+TN}{TP+FP+FN+TN} \tag{7}$$

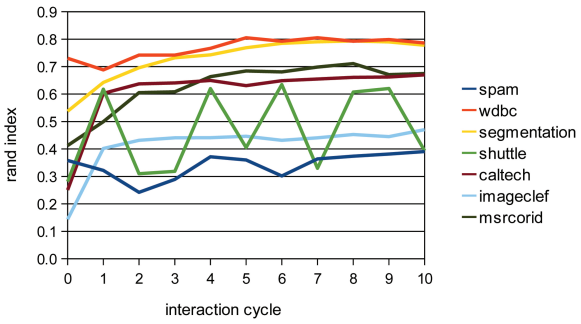

Figure 4: *Clustering using k-means, rand index values increase considerably after 10 interaction cycles.*

where TP is the number of instances belonging to a same class and assigned to the same class, TN is the number of instances in distinct classes assigned to distinct clusters, FP is the number of instances belonging to a same class but assigned to distinct clusters, and FN is the number of instances belonging to distinct classes but assigned to the same cluster. Lower values indicate worse clustering quality.

Table 4 shows *RI* values obtained before and after interactions. Since the seeds used to start k-means affect directly the resulting clusters (we are picking-up random seeds), we run k-means 120 times after each interaction step, getting the best *RI* value as output. Notice that *RI* values increase considerably just after three cycles of interactions for most datasets, over 250% for the *caltech* and 300% for the *imageclef*. Figure 4 shows *RI* values after 10 cycles of interactions, making clear the improvement for most datasets. The only exceptions are the *spam* and *shuttle* datasets. The reason why *RI* values for *spam* do not get better is that clusters are reasonably ill defined in that dataset. The *shuttle* dataset does not have instances evenly distributed among clusters. In fact, one of the clusters in *shuttle* contains approximately 80% of the data and, since k-means implicitly assumes a Gaussian distribution with unitary variance for the clusters, it does not perform well when this hypothesis is drastically violated. This assertion is confirmed by the silhouette coefficient, which indicates that the groups cohesion and separation of *shuttle* dataset really get better after iterations (see Figure 2).

4.1. Application

Taking advantage of the capability to improve the arrangement of instances in feature spaces we devise a visualization-assisted interactive application for content based image retrieval. Figure 5 outlines the reasoning behind the proposed system. Initially, target images are selected by the user and projected to the visual space together with a subset of representative images automatically selected from the image col-

Figure 5: *Overview of the image retrieval system. Initially the user defines the target images (highlighted with a red border). After that, a projection is created containing those images and a small sample drawn from the collection. The user can then reorganize the layout to create groups of similar images, which is used to transform the feature space. Finally, the most similar images to the target ones are retrieved.*

Table 4: *Rand index (accuracy) of k-means clustering of the original dataset and the results obtained varying the number of samples in each interaction cycle.*

Dataset	original	$\sqrt{n}/2$	$\sqrt{n}$	$3\sqrt{n}/2$
spam	0.3580	0.3220	0.2421	0.2891
wdbc	0.7302	0.6879	0.7421	0.7419
segmentation	0.5375	0.6424	0.6958	0.7322
shuttle	0.2812	0.6177	0.3096	0.3185
caltech	0.2506	0.6024	0.6368	0.6406
imageclef	0.1435	0.4018	0.4315	0.4406
msrcorid	0.4126	0.4993	0.6049	0.6083

lection. The user can then re-organize the layout moving images around to compose, for each target image, a group of similar images according to the user point of view. Images that are not similar to any target image can be grouped in a region of the image space to make up an "undesired" group of images. The modified layout is used to define transformations that will modify the dataset according to the new similarity relations. After transformation, for each target image a list containing the k most similar images is returned.

Figure 5 presents the result of applying our approach on the *caltech* image collection. In this example we seek for similar images considering two target samples, a motorcycle and a house. The target images are highlighted by a red border i n all steps. Notice that in the initial projection the target images are shuffled with images that are not of interest. After user manipulation, three groups are formed, one for each image of interest and a group representing the remaining irrelevant images. The 25 most similar images are returned for each target image. For the house, only 1 wrong image was retrieved and 3 for the motorcycle. For the sake of comparison Figure 6 presents the result of run the same search directly on the original feature space. Notice that 7 and 17 irrelevant images are returned for the house and motorcycle query images, respectively. Therefore, the precision ramps up from 72% to 96% for the motorcyle and from 32% to 88% for the house, attesting the effectiveness of our approach. Figure 7 presents the precision versus recall plot when the number of recovered images increases. The precision obtained with the transformed space is higher and stabilizes in high levels when more images are recovered while it decreases when images are directly retrieved from the original space.

Notice that if the coordinates of representative samples in the transformed space are also stored in the original space, new images added to the data collection can be transformed

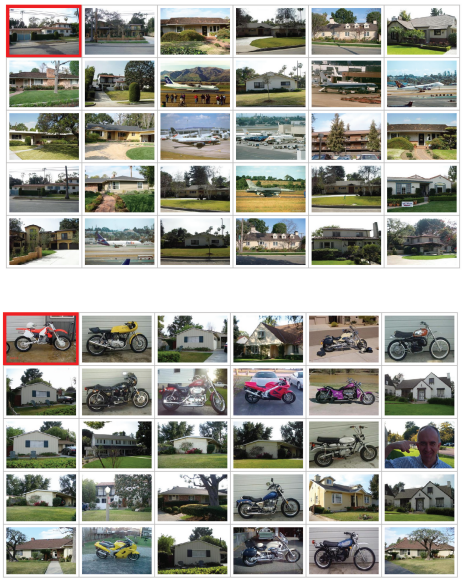

Figure 6: *Images retrieved using the original feature space. The precision worsen considerably if compared with the transformed space.*

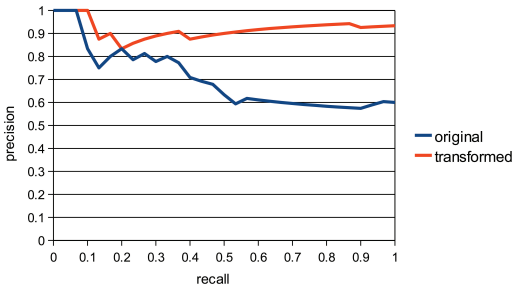

Figure 7: *Precision versus recall plot considering the original and transformed space.*

in a straightforward way by computing their affine transformation. Therefore, lists of similar images can quickly be updated as new images are added to the collection without demanding any user intervention. In fact, storing transformed attributes only for representative samples is enough to redefine the lists.

5. Discussion and Limitations

The tests presented in Section 4 clearly show the effectiveness of our approach to improve cohesion and separation between groups as well as to increase accuracy in classification and clustering of multidimensional datasets. Enabling user with simple and intuitive interactive tools for assisting group formation and classification is very desirable in many applications. Such a functionality is not easily found in current visualization-assisted feature space exploration systems, rendering the proposed approach an interesting alternative.

The need of a small number of representative samples (if compared to the total number of instances in the dataset) to transform the whole dataset is another attractive property of our approach. Although we are using $\sqrt{n}/2$ points at the start, our experiments show that this number can be much smaller if few groups are present in the dataset and representatives are uniformly picked out from each group. The connection between the number of representatives to be handled and the number of groups in the dataset is an intriguing issue that needs to be further investigated. Moreover, deciding which instances are more appropriate to be used as samples is also an issue to be investigated.

Another interesting aspect of our technique is that it is inherently incremental, that is, only the coordinates of representative samples in the transformed space need to be stored in order to modify the instances. This fact renders our approach appropriate to applications involving classification and clustering of streaming data.

Our experiments have shown that the proposed technique could not properly group instances that do not relate to each other. In fact, our tools turn out to be more efficient to improve existing groups than create new ones. This behavior becomes more evident when all representative samples are used to build the local mappings. In this case, the inherent global relationship of the data is incorporated into the local mapping, avoiding discontinuities and making changes less stringent.

The system prototype we have developed to show the effectiveness of our formulation has not been optimized for performance. The system was implemented in Java, including the eigen solver employed in Equation (3). Despite the lack of optimization, running times are very compelling. Considering the datasets presented on table 1, each cycle of interaction took on average 1.93s in an Intel(R) Core(TM)Duo 2.2GHz with 3GB of RAM.

6. Conclusion

In this paper we have proposed a novel approach for feature space transformation based on user manipulation of representative samples. Representative samples are mapped to the visual space via a multidimensional projection. The projected data can then be manipulated to create groups of interest from which local transformations are defined. Our experiments have shown that the proposed approach improves considerably the cohesion and separation of groups as well as the accuracy of classification methods.

The proposed methodology aims at adding the user on the loop of data analysis, classification and clustering without overwhelming him/her with complex interactive interfaces.

Using this premise we implement a simple and intuitive interactive system for image retrieval that is simple and intuitive to use.

We are currently investigating an extension of our methodology to handle streaming data. Detecting the appearance of new groups and selecting representative samples from those groups are challenge tasks we are also investigating.

Acknowledgments

The authors acknowledge the financial support of the Brazilian financial agencies CNPq and FAPESP.

References

[BPFG11] BERGER W., PIRINGER H., FILZMOSER P., GRÖLLER E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum 30*, 3 (2011), 911–920. 3

[CL11] CHANG C.-C., LIN C.-J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. 5

[CST00] CRISTIANINI N., SHAWE-TAYLOR J.: *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000. 5

[FA10] FRANK A., ASUNCION A.: UCI machine learning repository (http://archive.ics.uci.edu/ml), 2010. 4

[FHOM09] FERRI C., HERNÁNDEZ-ORALLO J., MODROIU R.: An experimental comparison of performance measures for classification. *Pattern Recognition Letters 30*, 1 (2009), 27 – 38. 5

[FPZ03] FERGUS R., PERONA P., ZISSERMAN A.: Object class recognition by unsupervised scale-invariant learning. In *CVPR* (2003), vol. 2, pp. 264–271. 4

[GWR09] GUO Z., WARD M. O., RUNDENSTEINER E. A.: Model space visualization for multivariate linear trend discovery. In *IEEE Symposium on Visual Analytics Science and Technology, VAST* (2009), pp. 75–82. 3

[HA85] HUBERT L., ARABIE P.: Comparing partitions. *Journal of Classification 2* (1985), 193–218. 6

[HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov. 2009), 1531–1538. 2

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. *VIS'90: Proceedings of the 1st Conference on Visualization* (1990), 361–378. 2

[JCC*11] JOIA P., COIMBRA D., CUMINATO J. A., PAULOVICH F. V., NONATO L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17* (Dec. 2011), 2563–2571. 2, 4

[JJ09] JOHANSSON S., JOHANSSON J.: Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov. 2009), 993–1000. 2

[JZF*09] JEONG D. H., ZIEMKIEWICZ C., FISHER B., RIBARSKY W., CHANG R.: ipca: An interactive system for pca-based visual analytics. *Computer Graphics Forum 28*, 3 (2009), 767–774. 3

[Kan00] KANDOGAN E.: Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *In Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics* (2000), pp. 9–12. 2

[LLL88] LOH H.-H., LEU J.-G., LUO R.: The analysis of natural textures using run length features. *Industrial Electronics, IEEE Transactions on 35*, 2 (may 1988), 323 –328. 4

[Mac67] MACQUEEN J. B.: Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), Cam L. M. L., Neyman J., (Eds.), vol. 1, University of California Press, pp. 281–297. 6

[MBN02] MOLINA L. C., BELANCHE L., NEBOT A.: Feature selection algorithms: A survey and experimental evaluation. In *Proceedings of the 2002 IEEE International Conference on Data Mining* (Washington, DC, USA, 2002), ICDM '02, IEEE Computer Society, pp. 306–. 2

[Mic12] MICROSOFT RESEARCH CAMBRIDGE: Object recognition image database (http://research.microsoft.com/), 2012. 4

[PEP*11] PAULOVICH F. V., ELER D. M., POCO J., BOTHA C. P., MINGHIM R., NONATO L. G.: Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum 30*, 3 (2011), 1091–1100. 2

[R C12] R CORE TEAM: R: A language and environment for statistical computing. ISBN 3-900051-07-0. 5

[Rou87] ROUSSEEUW P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math. 20*, 1 (1987), 53–65. 5

[SS05] SEO J., SHNEIDERMAN B.: A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization 4*, 2 (July 2005), 96–113. 2

[TFH11] TURKAY C., FILZMOSER P., HAUSER H.: Brushing dimensions: A dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2591–2599. 2

[TMN03] TEJADA E., MINGHIM R., NONATO L. G.: On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization 2*, 4 (2003), 218–231. 3, 4

[YHW*07] YANG J., HUBBALL D., WARD M. O., RUNDENSTEINER E. A., RIBARSKY W.: Value and relation display: Interactive visual exploration of large data sets with hundreds of dimensions. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (May 2007), 494–507. 2

[YWRH03] YANG J., WARD M. O., RUNDENSTEINER E. A., HUANG S.: Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings of the symposium on Data visualisation 2003* (Aire-la-Ville, Switzerland, Switzerland, 2003), VISSYM '03, Eurographics Association, pp. 19–28. 2