EuroVis 2013 Workshop on Visual Analytics using Multidimensional Projections (2013)
M. Aupetit and L. van der Maaten (Editors)

A Proposition of Interactive Visual Clustering System

P. Bruneau and B. Otjacques

CRP - Gabriel Lippmann, Belvaux, Luxembourg

Abstract

This work describes a novel interactive visual clustering system. It combines a 2D projection with a clustering algorithm that operates on this projected data. Example-based interactions are supported directly through the 2D representation. Each interaction incrementally updates the 2D projection and the associated clustering.

Categories and Subject Descriptors (according to ACM CCS): H.5.m [Information Storage an Retrieval]: Information Interfaces and Presentation—Miscellaneous

1. Introduction

Clustering is a prevalent task for understanding, and summarizing complex data. This approach is usually taken exploratively, when we do not have any explicit prior knowledge about the data.

Real data sets are often high dimensional. Setting up a visual clustering system is thus not trivial, and depends on the existence of adequate low dimensional representations (preferably 2D). Rather independently of work on the clustering topic, the projection of high-dimensional data in a 2D space has been thoroughly investigated. Using this range of techniques, the data becomes affordable for interaction.

We advocate the projection of the data and its clustering in the same 2D view. The originality of this work lies in an interactive loop, that allows the user to influence the clustering result directly through the 2D visualization. More specifically, we support an input based on examples, where the user can provide his expectations regarding pairs of elements in the 2D projection. Other views of the data (e.g. inspector) may complement our system, and suggest alternative similarity and clustering patterns to users. Ultimately, preferences of users with respect to the distribution of the data in the projection space are expected to influence the clustering structure, e.g. tend to regroup originally dissimilar clusters. The difficulty of this approach lies in an elegant combination of this subjective supervision, and the intrinsic nature of the data.

In this paper, we operate on data sets through similarity matrices. We hypothesise that user interaction may be elegantly handled by influencing these matrices. In this context, kernel-based methods seemed an obvious choice to ground our work. They focus on processing positive semi-definite similarity matrices (*kernel* matrices), and were successfully applied to the problems of projecting data in low-dimensional spaces (kernel PCA projection) and clustering (spectral clustering algorithms).

We motivate our kernel transformation with a detailed analysis, and the care of optimizing the effect of user interactions: when a user specifies as few as one or two constraints, his actions should lead to a tangible feedback on the visualization, and the current clustering.

After a review of the related work, and an overview of the targeted system, we give a detailed description of our interactive loop. Specifically, the translation of user interactions into binary relations is formalised. To maximise the cover of the relations, and thus the area of influence of user interactions, we derive and justify the use of transitive closures. Pairwise similarities associated to members of these relations are transformed by adapted functions; some insight to the desirable properties of such functions is given, and supports our eventual choice. We then conclude this paper with references to preliminary experiments, and perspectives for future work.

2. Related Work

2D projections are a common way to represent high-dimensional numerical data. PCA is probably the most popular technique in this range. It seeks the linear subspace that captures the maximal variance from the data. Its good interpretability taken aside, this method tends to compress the elements in the projection space (i.e. on average, the normalised pairwise distances are smaller in the projection

DOI: 10.2312/PE.VAMP.VAMP2013.001-004

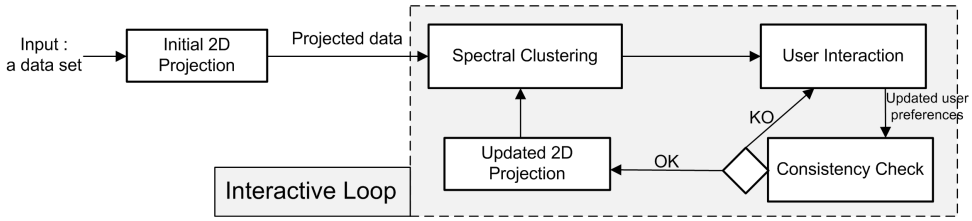

Figure 1: *Description of the envisioned interactive visual clustering system.*

space than in the original space). *Self Organizing Maps* (SOM) and *Multi Dimensional Scaling* (MDS) are other popular methods in this domain (see [Aup07] for a more extensive review).

Kernel PCA [SSM98] is somehow affiliated to MDS, as it resorts to the eigen-decomposition of a kernel similarity matrix (e.g. computed using the data in the original space). This method can be seen as a linear projection on the 2D principal non-linear manifold that underlies the similarity matrix. For details and insight about kernel PCA, the interested reader may refer to [SSM98, Bru13].

The projection of d-dimensional $(d > 2)$ data to a 2-dimensional space inevitably leads to some information loss, materialised by projection artefacts, i.e. distortions induced by the transformed 2D space with respect to the distribution of pairwise distances. The reader may consult [Aup07] for a review on this matter. In brief, compression (respectively stretching) occurs when the normalised pairwise distances in the projected space are smaller (respectively greater) than their counterpart in the original space. The typical distortions associated to the kernel PCA 2D projection have already been shortly discussed [Bru13]. Even if not a primary concern in this paper, projection artefacts reflect how influential a transformation may be, and will be measured and discussed in our experimental section.

The objective of visual clustering is to go beyond an effective representation, and also allow a level of interaction. The implicit goal, and expected benefit, is to allow a user to gain more insight to his data, and clustering algorithm, through intuitive manipulations. For example, in [AAR*09], in the context of spatio-temporal data clustering, the parameters of the clustering algorithm are adjustable in the user interface, with visual feedback on the implied clustering result. In contrast, the present paper uses a non-parametric approach, where users can provide examples through element selection.

We propose a principled approach to convey user interactions as a similarity matrix transformation. This transformed matrix is then processed by a standard spectral clustering algorithm [NJW01]. Indeed, projecting data according to its similarity matrix may lead to clusters with arbitrary shapes, and the spectral approach is especially adapted to this case. Interestingly, the latter work highlights the intricate relationship between kernel PCA and spectral clustering. Actually, the methods mostly differ on the employed normalisations. The ability of kernel-based methods to handle both visualisation and clustering motivated our choice: both facets thus integrate naturally in a unified formalism.

3. Method Description

This work ultimately aims at implementing the interactive clustering system sketched in Figure 1. This section is dedicated to the core of our contribution: the interactive loop in Figure 1.

3.1. From user interactions to binary relations

To support the discussion, let us define a data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with each $\mathbf{x} \in \mathbf{X}$ taking values in $\mathbb{R}^d$. We assume the existence of a $N \times N$ similarity matrix $\mathbf{K}$, with values in $[0,1]$, such that $\mathbf{K}_{ij} = \text{similarity}(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{K}_{ii} = 1$. Specifying pairs of elements that should be closer (*link*) or further (*not-link*) from each other is natural for users, and requires few prior information about the data distribution. Our intuition is to guide the clustering process by transforming the projected space it operates on: to do so, user preferences have to be translated into a transformation of the 2D projection. The first step in this direction is to formalise user inputs in terms of binary relations.

Let us define the *Link* (respectively *Not-Link*) symmetric, irreflexive binary relation $\mathcal{L}$ (respectively $\mathcal{N}$), so that:

$$\mathbf{x} \text{ and } \mathbf{x}' \text{ are linked} \Leftrightarrow \mathbf{x}\mathcal{L}\mathbf{x}'$$
$$\mathbf{x} \text{ and } \mathbf{x}' \text{ are not linked} \Leftrightarrow \mathbf{x}\mathcal{N}\mathbf{x}'$$

The intersection between $\mathcal{L}$ and $\mathcal{N}$ is constrained to be empty. In Figure 2a, we illustrate, with a toy example, how few user-specified constraints translate into instances of these relations.

As we consider linking constraints, two induction rules seem rather intuitive in this context:

$$\mathbf{x}_i\mathcal{L}\mathbf{x}_j \text{ and } \mathbf{x}_j\mathcal{L}\mathbf{x}_k \Rightarrow \mathbf{x}_i\mathcal{L}\mathbf{x}_k, \quad (1)$$
$$\mathbf{x}_i\mathcal{L}\mathbf{x}_j \text{ and } \mathbf{x}_j\mathcal{N}\mathbf{x}_k \Rightarrow \mathbf{x}_i\mathcal{N}\mathbf{x}_k. \quad (2)$$

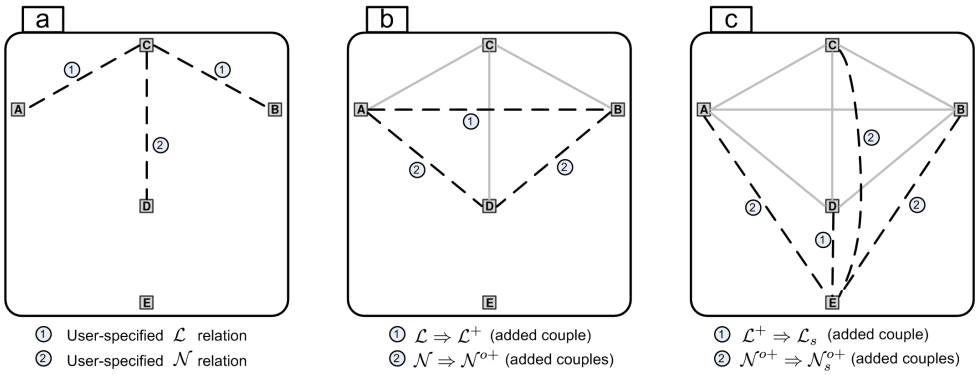

Figure 2: ***a)*** *User interactions are formalised in the relations.* ***b)*** *These relations are extended by their closures.* ***c)*** *The cover is increased by the similarity-augmented relations.*

Examples of inductions according to (1) and (2) are given in Figure 2b. Let us note that the augmentation of $\mathcal{L}$ by rule (1) is $\mathcal{L}^+$, the transitive closure of $\mathcal{L}$. Rule (2) cannot be expressed in standard binary relations terminology. As the notions of composition and transitivity intervene, we coin this operation as the *composite transitivity* of $\mathcal{L}$ and $\mathcal{N}$, and symbolise it by $\mathcal{N}^{o+}$.

We also constrain $\mathcal{L}^+ \cap \mathcal{N}^{o+} = \emptyset$. In the context of our interactive clustering system (see Figure 1), $\mathcal{L}$ and $\mathcal{N}$ are used to record the user interactions. (1) and (2) are not intended to replace these: a user has to keep an easy track of his actual past interactions. Rather, they are used to check the consistency of the current $\mathcal{L}$ and $\mathcal{N}$. More specifically, just after an interaction, if we have $\mathcal{L}^+ \cap \mathcal{N}^{o+} \neq \emptyset$, the current $\mathcal{L}$ and $\mathcal{N}$ are said to be *inconsistent*: the user is then asked to revise his past interactions (see "consistency check" in Figure 1). If the consistency is verified, the pairwise similarities associated to couples lying in the closures are modified.

The closures $\mathcal{L}^+$ and $\mathcal{N}^{o+}$ are exemplified in Figure 2b. In practice, when several hundred data points lie in our interactive clustering system, we do not expect a user to perform more than 5 or 10 interactions: consequently, even after applying inductions, relations $\mathcal{L}^+$ and $\mathcal{N}^{o+}$ are likely to be very sparse.

We propose to use the similarities between elements to augment the cover of $\mathbf{X}$ by $\mathcal{L}^+$ and $\mathcal{N}^{o+}$. To this aim, we define the *one-sided restriction* of a symmetric relation as:

$$\mathbf{X}|_{\mathcal{R}} = \{\mathbf{x} \in \mathbf{X} | \exists \mathbf{x}' \in \mathbf{X}, \mathbf{x}\mathcal{R}\mathbf{x}'\}$$

Intuitively, it seems natural that all elements that are neither in $\mathbf{X}|_{\mathcal{L}}$ nor in $\mathbf{X}|_{\mathcal{N}}$ can be artificially linked to their most similar element found in the restrictions. This may be seen as a k-NN step, with $k = 1$. Formally, the similarity-augmented *link* relation $\mathcal{L}_s$ is derived as follows:

$$\mathbf{x}_i \mathcal{L}_s \mathbf{x}_j \Leftrightarrow \begin{cases} \Big(\mathbf{x}_i \notin \mathbf{X}|_{\mathcal{L}^+ \cup \mathcal{N}^{o+}} \\ \quad \text{and } j = \arg\max_{j | \mathbf{x}_j \in \mathbf{X}|_{\mathcal{L}^+ \cup \mathcal{N}^{o+}}} \mathbf{K}_{ij} \Big) \\ \text{or } \mathbf{x}_i \mathcal{L}^+ \mathbf{x}_j \end{cases}$$

The closures $\mathcal{L}_s^+$ and $\mathcal{N}_s^{o+}$ follow mechanically from applying (1) and (2) to $\mathcal{L}_s$ and $\mathcal{N}$, and can also be used for consistency checks. These similarity-augmented relations are illustrated by Figure 2c.

Yet, such checks are now useless: the consistency of $\mathcal{L}$ and $\mathcal{N}$ implies that of $\mathcal{L}_s$ and $\mathcal{N}$. The proof can easily be sketched: let us consider the $\mathbf{x}_j$ selected by the first proposition on the right hand side of (3). During a subsequent consistency check of $\mathcal{L}_s$ and $\mathcal{N}$, any instantiation implying one of these $\mathbf{x}_j$ can match either (1) or (2), but not both at the same time.

3.2. Transforming the kernel similarity matrix

In the previous section, we formalized the recorded user interactions. Implicit augmentations of the baseline relations were also discussed. Our further intuition may be then summarised as follows: *have linked elements more similar, and not linked elements more dissimilar*.

Formally, functions that implement this intuition have to be determined. We propose to use the two following function families, for application to similarity values in $[0,1]$:

$$f^{\alpha}_{\text{link}}(\text{sim}) = \text{sim}^{\frac{1}{\alpha}} \quad (3)$$

$$f^{\alpha}_{\text{not-link}}(\text{sim}) = 1 - (1 - \text{sim})^{\frac{1}{\alpha}} \quad (4)$$

with f^{α}_{link} (respectively $f^{\alpha}_{\text{not-link}}$) the family of functions

that tends to augment (respectively diminish) the parameterised similarity. Let us remark that a similarity matrix fully or partly transformed by these functions remains a valid similarity matrix, as the image of $[0,1]$ by these functions is also $[0,1]$.

This choice obeys the following desirable properties:

- Elements that must be linked, and are already close do not need further similarity increase. Linked elements with low similarity must be more strongly influenced.
- Reciprocally, close elements that must not be linked need a strong influence, purposely to create an artificial boundary. Couples in $\mathcal{N}$ that are already dissimilar should not be much influenced.
- If a couple in $\mathcal{N}$ (respectively in $\mathcal{L}$) is extremely similar (respectively dissimilar), trying to separate (respectively regroup) it would tear the whole projection apart: below some threshold, the influence is thus softened. Such violations of user preferences might be highlighted with a color code.

These properties were already illustrated with instances of f^{α}_{link} and $f^{\alpha}_{\text{not-link}}$ [BO13].

Eventually, f^{α}_{link} (respectively $f^{\alpha}_{\text{not-link}}$) is applied to pairwise similarities of couples lying in a *link* (respectively *not-link*) relation. The application to simple closures (i.e. $\mathcal{L}^{+}$ and $\mathcal{N}^{o+}$), and to similarity augmented closures (i.e. $\mathcal{L}^{+}_{s}$ and $\mathcal{N}^{o+}_{s}$) was evaluated with α empirically set to 6 [BO13].

$f^{\alpha}_{\text{not-link}}$ was chosen for reasons of symmetry with f^{α}_{link}. Another candidate would intuitively have been the sim^{α} family. However, the maximal magnitude of the influence of the latter family of functions tends to occur close to 0.5 [BO13]. Consequently, similar couples lying in $\mathcal{N}$ would not be sufficiently separated.

So far, we have not defined how the similarity values in $\mathbf{K}$ are computed. In the context of kernel methods, this is achieved through the use of a *kernel function* parameterised by a couple of elements given in the original data space. The Gaussian kernel function $\mathbf{k}_{\text{Gauss}}$ is a typical choice in this context [Bru13].

The effectiveness of the function families defined by (3) and (4) is somehow conditioned by the uniformity in $[0,1]$ of the similarity values in $\mathbf{K}$. However, similarity values distributions, as generated by the Gaussian kernel function, may be data and dimensionality dependent, and far from uniformity [Bru13]. Thus, we rather compute the similarity matrix $\mathbf{K}$ with the p-Gaussian kernel function $\mathbf{k}_{\text{pGauss}}$ (see [FWV05, Bru13] for details). Let us remark that unlike the widely employed Gaussian kernel, the p-Gaussian does not lead to positive semi-definite kernel matrices [Bru13]: this may be an issue for some kernel-based methods, such as SVM classifiers. Yet, kernel PCA only requires the two major eigenvalues, which are positive, when using $\mathbf{k}_{\text{pGauss}}$, for all but extremely degenerate data distributions [Bru13]. $\mathbf{k}_{\text{pGauss}}$ also leads to more stable eigen-decompositions, which supports the robustness of our system.

4. Conclusion

In this paper, we proposed and described an interactive visual clustering system, that grounds on a 2D projection of numerical data sets. Users have the possibility to provide their preference to the system, by indicating pairs of elements they would like to see close or far apart. The 2D projection, and subsequently the clustering that operates on it, incorporates this interaction in a compromise with baseline similarities.

Preliminary results from batch experiments [BO13] show the effectiveness of the method, with as few as one interaction leading to tangible influence on the visualisation, and subsequent clustering. Yet, due to a random sampling scheme, the resulting clustering quality can be seen as a lower bound on what could be expected. An actual implementation of the interactive loop, along with subjective user tests, would refine the assessment of our system.

Beyond naively associating distinguishable shapes and colours to data points according to their cluster memberships, it would be desirable to associate a global shape to each cluster. Yet, spectral approaches do not assume a specific cluster shape (whereas e.g. elliptic shapes for Gaussian mixture based algorithms). A possibility would be to adapt work on *blob* construction, where arbitrary shapes are built from density analysis [SBG00].

References

[AAR*09] ANDRIENKO G., ANDRIENKO N., RINZIVILLO S., NANNI M., PEDRESCHI D., GIANOTTI F.: Interactive visual clustering of large collections of trajectories. *IEEE VAST* (2009), 3–10. 2

[Aup07] AUPETIT M.: Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing* (2007), 1304–1330. 2

[BO13] BRUNEAU P., OTJACQUES B.: *An Interactive, Example-Based, Visual Clustering System*. Tech. Rep. hal-00797367, CRP - Gabriel Lippmann - Department of Informatics, 2013. 4

[Bru13] BRUNEAU P.: *On the visualization of high-dimensional data*. Tech. Rep. hal-00787488, CRP - Gabriel Lippmann - Department of Informatics, 2013. 2, 4

[FWV05] FRANÇOIS D., WERTZ V., VERLEYSEN M.: About the locality of kernels in high-dimensional spaces. *ASMDA* (2005), 238–245. 4

[NJW01] NG A. Y., JORDAN M. I., WEISS Y.: On spectral clustering: Analysis and an algorithm. *NIPS* (2001). 2

[SBG00] SPRENGER T. C., BRUNELLA R., GROSS M. H.: H-BLOB: a hierarchical visual clustering method using implicit surfaces. *IEEE VIS* (2000), 61–68. 4

[SSM98] SCHÖLKOPF B., SMOLA A., MÜLLER K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* (1998), 1299–1319. 2