An Interactive, Example-Based, Visual Clustering System

Pierrick Bruneau, Benoît Otjacques

An Interactive, Example-Based, Visual Clustering System

Pierrick Bruneau [*1] and Benoit Otjacques [†1]

[1]CRP - Gabriel Lippmann, Department of Informatics
41, rue du Brill, L-4422 Belvaux (Luxembourg)

Abstract

This work describes and evaluates a novel interactive visual clustering system. It combines a 2D projection with a clustering algorithm that operates on this projected data. Users can interact directly through the 2D representation, by providing examples according to their expert ground truth. Each interaction incrementally updates the 2D projection and the associated clustering. Experiments show the effectiveness of the method, with as few as one interaction leading to a tangible influence on the visualization.

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

[*] bruneau@lippmann.lu
[†] otjacque@lippmann.lu

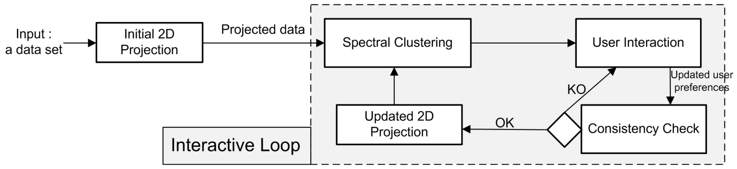

Figure 1: Description of the envisioned interactive visual clustering system.

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. The complexity of our method is discussed, and the results of our experimental evaluation are presented. Observations are drawn, and supported by a statistical analysis. We then conclude this paper with some 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 space than in the original space). *Self Organizing Maps* (SOM) and *Multi Dimensional Scaling* (MDS) are other popular methods in this domain (see [2] for a more extensive review).

Kernel PCA [11] 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 [11, 3].

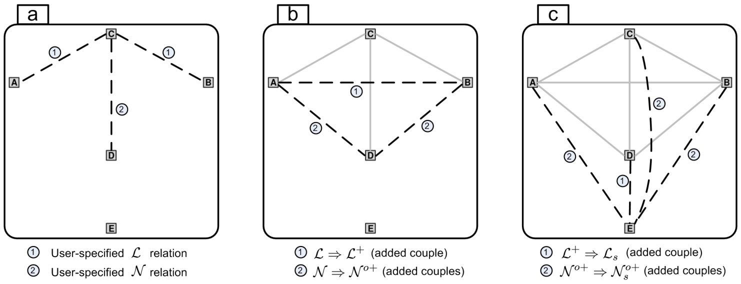
a
b
c
A
B
C
D
E
① User-specified $\mathcal{L}$ relation
② User-specified $\mathcal{N}$ relation
① $\mathcal{L} \Rightarrow \mathcal{L}^+$ (added couple)
② $\mathcal{N} \Rightarrow \mathcal{N}^{o+}$ (added couples)
① $\mathcal{L}^+ \Rightarrow \mathcal{L}_s$ (added couple)
② $\mathcal{N}^{o+} \Rightarrow \mathcal{N}_s^{o+}$ (added couples)

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.

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 [2] 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 [3]. 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 present work is loosely related with semi-supervised clustering approaches (e.g. [4]). These classically convert the user interactions to clustering constraints (i.e. pre-labelled elements, or pairwise constraints), that are incorporated in an explicit objective function. Alternatively, we rather 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 [9]. 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.

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 [1], 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.

[10] is closely related to our work, and proposes to let the user position an automatically selected sample of data elements in the 2D projected space. The rest of

the collection is then arranged in the 2D space as a compromise between user preference and data similarity. Their technique uses a combination of clustering and SVM classifiers.

3 Method Description

This work ultimately aims at implementing the interactive clustering system sketched in Figure 1. The initial 2D projection phase is determined by computing a kernel PCA on the data in its original space. The data is clustered in the projected 2D space using a spectral clustering algorithm. As evoked above, this choice is rather natural when operating on data resulting from a kernel PCA 2D projection. The rest of the 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:

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

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, some degree of transitivity seems intuitive. We thus define the two following closures:

$$\mathbf{x}_i\mathcal{L}^+\mathbf{x}_k \Leftrightarrow \begin{cases}\mathbf{x}_i\mathcal{L}\mathbf{x}_k \\ \text{or } \exists j \text{ so that } \mathbf{x}_i\mathcal{L}\mathbf{x}_j \text{ and } \mathbf{x}_j\mathcal{L}\mathbf{x}_k,\end{cases} \tag{1}$$

$$\mathbf{x}_i\mathcal{N}^{o+}\mathbf{x}_k \Leftrightarrow \begin{cases}\mathbf{x}_i\mathcal{N}\mathbf{x}_k \\ \text{or } \exists j \text{ so that } \mathbf{x}_i\mathcal{L}\mathbf{x}_j \text{ and } \mathbf{x}_j\mathcal{N}\mathbf{x}_k.\end{cases} \tag{2}$$

$\mathcal{L}^+$ is exactly the transitive closure of $\mathcal{L}$. $\mathcal{N}^{o+}$ cannot be expressed in standard binary relations terminology. As the notions of composition and transitivity intervene, we coin this relation as the *composite transitive closure* of $\mathcal{L}$ and $\mathcal{N}$.

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)

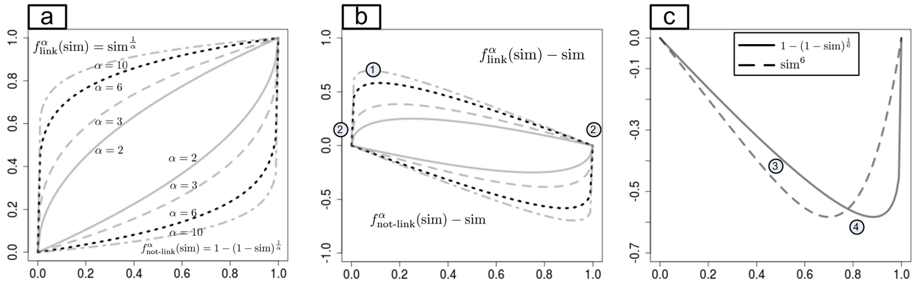

Figure 3: **a)** Instances of the families of functions. **b)** To highlight the properties of these families, their difference to the similarities they apply to is also plotted. **c)** The two candidates for $f^{\alpha}_{\text{not-link}}$ are highlighted. **1)** The maximal influence of the transformations occurs close to extreme similarty values (0.1 or 0.9). **2)** The families are continuous at the similarity bounds. **3)** The chosen candidate has a weaker slope, for a smoother influence on the similarity distribution. **4)** The location of the maximal influence better reflects the desired properties for our transformation.

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 $\mathcal{N}_s^{o+}$ are then defined respectively as the transitive closure of $\mathcal{L}_s$, and the composite transitive closure of $\mathcal{L}_s$ and $\mathcal{N}$. They are built by replacing $\mathcal{L}$ by $\mathcal{L}_s$ respectively in Equation (1) and (2).

These new closures may also be checked for consistency. 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 $[0, 1]$:

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

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

with f^{α}_{link} (respectively $f^{\alpha}_{\text{not-link}}$) the family of functions that tends to augment (respectively diminish) the parameterised similarity. Examples of these functions are represented in Figure 3a for several values of α. 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 $[0, 1]$.

Figure 3b illustrates the motivations that led us to these monotonic and smooth functional forms:

- 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.

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 $\mathcal{N}_s^{o+}$), will both be tested in our experimental section. Throughout the rest of the paper, α is empirically set to 6.

$f^{\alpha}_{\text{not-link}}$ was chosen for reasons of symmetry with f^{α}_{link}. Another candidate would intuitively have been the sim^{α} family. However, as plotted in Figure 3c, the maximal magnitude of the influence of the latter family of functions tends to occur close to 0.5. 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 [3].

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 [3]. Thus, we rather compute the similarity matrix $\mathbf{K}$ with the p-Gaussian kernel function $\mathbf{k}_{\text{pGauss}}$ (see [7, 3] for details). Let us remark that unlike the widely employed Gaussian kernel, the p-Gaussian does not lead to positive semi-definite kernel matrices [3]: 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 [3]. For example, the p-Gaussian kernel matrix computed on a sample of 400 points from the *Unidat-10* data set (see section 5 for a data set description) has approximately 50% of positive eigenvalues. This amount is not much changed when considering the similarity augmented transformation (10 random couples in $\mathcal{L}$ and $\mathcal{N}$). Moreover, the spectral clustering algorithm (see Figure 1) recomputes its own kernel matrix by analysing the 2D projected data. In this work, $\mathbf{k}_{\text{pGauss}}$ is thus safe to use and transform. $\mathbf{k}_{\text{pGauss}}$ also leads to more stable eigen-decompositions [3], which supports the robustness of our system.

4 Complexity

The computation of the baseline similarity matrix $\mathbf{K}$ is $O(dn^2)$, but is only required once, at the system initialization. Each interaction leads to the modification of $\mathcal{L}$ or $\mathcal{N}$. A naive approach to the update of their closures (see section 3.1) would lead to $O(n^2)$ operations: yet, exploiting the incrementality of this construction ultimately permits a linear update cost.

The incompressible burden of the method is located in the eigen-decomposition of the similarity matrix, required afresh at each interaction. Due to its very quick convergence, and the restriction to the two major eigenpairs, the Iteratively Restarted Lanczos method can be considered as approximately $O(dn^2)$, which is in the same order of the pairwise similarity matrix computation. In practice, on a standard workstation, the computation is tractable (i.e. interaction taking less than 1 second) when handling up to a few thousand elements. Beyond this order of magnitude, the relevance of projecting such an amount of data points in a closed 2D space may be questioned, and research would rather be oriented towards combinations with hierarchical data structures (e.g. dendrogram). Cuts at specific heights would help control the displayed level of detail. The algorithmic consequences should be studied carefully, and may lead to adapted optimizations with a $O(dn'^2), n' < n$ cost at each step.

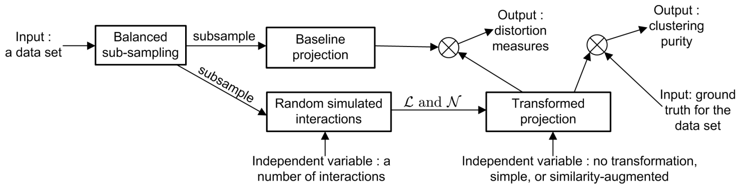
Input : a data set
Balanced sub-sampling
subsample
Baseline projection
subsample
Random simulated interactions
$\mathcal{L}$ and $\mathcal{N}$
Transformed projection
Output : distortion measures
Output : clustering purity
Input: ground truth for the data set
Independent variable : a number of interactions
Independent variable : no transformation, simple, or similarity-augmented

Figure 4: Description of our experimental protocol.

data set	**# attributes**	**# elements**	**# classes**
Pima	8	768	2
Isolet	617	1500	5
Unidat-10	10	2000	2
Unidat-500	500	2000	2

Table 1: Description of the data sets used in the experiments

5 Experiments

The actual system, as we envisioned it, is summarised in Figure 1. In this paper, we choose to focus on evaluating the proposed kernel transformation in itself, and its consequences with respect to metrics such as the intensity of projection artefacts, and the clustering purity: we thus propose an alternative, batch experimentation protocol. Doing so, we expect to evaluate our method more objectively. This protocol is overviewed in Figure 4.

For this evaluation, we used four numerical data sets, summarised in table 1. *Pima* and *Isolet* are classical UCI data sets. For *Isolet*, we selected the five classes associated to the vowels. All but one attributes of the *Unidat* sets were drawn from a uniform law in $[0, 1]$. The two classes of 1000 elements differ by the remaining attribute: it was drawn uniformly in $[0, 0.5]$ (respectively $[0.5, 1]$) for the first (respectively last) 1000 elements.

With this data set selection, we tried to cover the most classical difficulties encountered in machine learning. *Pima* is well known for its erroneous and missing data [6], that even state-of-the-art classifiers have some difficulty to process successfully. With respectively 500 and 617 dimensions, *Unidat-500* and *Isolet* are good representatives of high dimensional numeric data.

In Figure 4, we see that an experiment is characterised by a specific data set (and its respective ground truth), a number of interactions n_{int} for each relation $\mathcal{L}$ and $\mathcal{N}$, and a transformation method. We define a *control* method for the similarity matrix transformation, that simply keeps the baseline unchanged. To evaluate the importance of the similarity augmented approach, we define the *simple* (respectively *augmented*) method, that transforms the similarity matrix according to $\mathcal{L}^+$ and $\mathcal{N}^{o+}$ (respectively $\mathcal{L}^+$ and $\mathcal{N}^{o+}$).

We choose to perform each experiment independently 20 times, and record the relevant output, i.e. the medians of the compression and stretching artefacts (see [2] for computational details), and the purity of the clustering output by the spectral algorithm.

The latter is parameterised with the ground truth number of classes. As seen in Figure 4, to clearly measure the influence of the proposed interactions, the distortion artefacts are computed with respect to the baseline p-Gaussian kernel PCA 2D projection. For more robust results, a balanced subsample (i.e. that reflects the original class distribution) was drawn randomly and independently for each experiment (depending on the data set, $\simeq 300$ elements).

Are user interactions influential on distortion or clustering purity? Is the *simple* method sufficient? To answer these questions, we define n_{int} and the transformation method as independent variables of two-way independent ANOVA tests. Accounting for the amount of independent experiments we carried out, normality and homogeneous variances of the gathered metrics are not required [5].

Using any method induces significant distortion artefacts. We observe the same pattern for all data sets, and both artefacts (i.e. compression and stretching): slightly increasing linear trend with respect to n_{int} using the *simple* method (e.g. with *Pima*, from 0.01 to 0.04 on average), and stronger decreasing linear or quadratic trend with the *augmented* method (e.g. with *Isolet*, from 0.13 to 0.07 on average, $F(1, 342) = 58$, $p < 10^{-12}$).

The observations regarding the clustering purity are much more data set dependent. Both methods do not make a significant difference for *Pima* and *Unidat-500*. As an explanation, let us note again that *Pima* is especially difficult for classification tasks [6], and that *Unidat-500* is extremely noisy. For *Isolet* and *Unidat-10*, only the *augmented* method is influential (e.g. $F(1, 342) = 27$, $p < 10^{-6}$ for *Isolet*). Again, similar patterns are observed for both data sets: the purity is harmed when $n_{\text{int}} = 1$, but raises, and overcomes the baseline results with increased interaction. For *Unidat-10* and $n_{\text{int}} = 1$, with the *augmented* method the purity is 56.5%, instead of 62% on average for the *control* method. This metric then follows a significant linear trend ($F(1, 342) = 17$, $p < 10^{-4}$) to reach 66.5% purity for $n_{\text{int}} \geq 7$.

6 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.

From the experimental results, we see that our proposition, using the similarity-augmented method, leads to the expected result that as few as one interaction has a tangible effect on the 2D projection (reflected by increased distortions), and subsequent clustering. Further interactions, when made according to the ground truth classes, tend to smoothen this strong prior effect, with improved clustering results.

Due to our 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.

First tests using a working interactive prototype revealed some flaws about our scheme. Specifically, in some cases, interactions lead to exert the required influence,

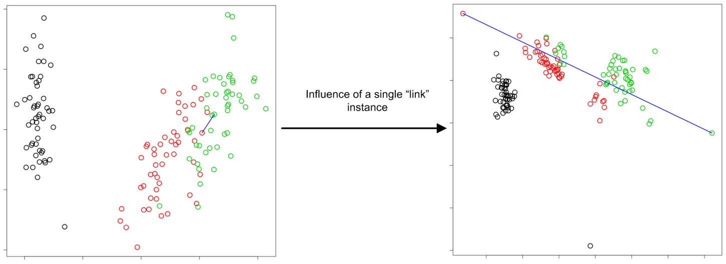

Figure 5: Example of influence of a single instance from $\mathcal{L}$. The resulting groups are satisfactory, but this tends to produce isolated points.

but isolate the interacted points far apart any other element in the projection. This seems to emerge from our transitive closure scheme, which omits to transfer the influence to all pairs of elements in the respective neighborhoods of a couple in a relation. This problem is illustrated in Figure 5.

We intend to correct this problem by the use of respective neighborhoods of all elements involved in a relation, and applying the transformation on all links in the complete bipartite graph between the neighborhoods of a given instance in $\mathcal{L}$ or $\mathcal{N}$. Another complementary perspective would be to limit neighborhoods using the information returned by the spectral clustering algorithm at the previous interactive step.

Alternatively to the p-Gaussian function, we also plan to evaluate the distance transformation method presented in [3]. This leads to the same theoretical difficulties regarding positive semi-definiteness, but would allow to apply a transformation function directly on the distances, rather than on the kernel values. This might lead to different results, and will be studied later on.

The kernel PCA projection is stable up to an affine transformation, i.e. small changes in the data may lead to important rotation/mirorring effects. This problem could be straghtforwardly handled using a Procrustes post-processing approach [8].

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 [12].

References

[1] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. *IEEE Symposium on Visual Analytics Science and Technology*, pages 3–10, 2009.

[2] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, pages 1304–1330, 2007.

[3] P. Bruneau. On the visualization of high-dimensional data. Technical report, CRP - Gabriel Lippmann - Department of Informatics, 2013.

[4] C. Domeniconi, J. Peng, and B. Yan. Composite kernels for semi-supervised clustering. *Knowledge and Information Systems*, 28(1):99–116, 2011.

[5] T. S. Donaldson. Robustness of the F-test to errors of both kinds and the correlation between the numerator and denominator of the F-ratio. *Journal of the American Statistical Association*, pages 660–676, 1968.

[6] A. Feelders. Handling missing data in trees: Surrogate splits or statistical imputation? *Principles of Data Mining and Knowledge Discovery, LNCS 1704*, pages 329–334, 1999.

[7] D. François, V. Wertz, and M. Verleysen. About the locality of kernels in high-dimensional spaces. *International Symposium on Applied Stochastic Models and Data Analysis*, pages 238–245, 2005.

[8] F. J. Garcia-Fernandez, M. Verleysen, J. A. Lee, and I. Diaz. Stability comparison of dimensionality reduction techniques attending to data and parameter variations. *EuroVis 2013 Workshop on Visual Analytics using Multidimensional Projections*, 2013.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Neural Information Processing Symposium*, 2001.

[10] J. Philippeau, J. Pinquier, P. Joly, and J. Carrive. Dynamic organization of audiovisual database using a user-defined similarity measure based on low-level features. *IEEE International Conference on Image Processing*, pages 33–36, 2008.

[11] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[12] T. C. Sprenger, R. Brunella, and M. H. Gross. H-blob: a hierarchical visual clustering method using implicit surfaces. *Proceedings of the Conference on Visualization*, pages 61–68, 2000.