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

Short Papers

User control for adjusting conflicting objectives in parameter-dependent visualization of data

E. Renard, P. Dupont and M. Verleysen

ICTEAM Institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Abstract

Dealing with high-dimensional data becomes very common nowadays; visualization is a natural preprocessing to have an overview of such data. A lot of dimensionality reduction methods exist; many of them require to tune a parameter implementing a trade-off between conflicting objectives. Automatically choosing the appropriate trade-off is usually a difficult task because in most cases the exact final goal of the visualization is ill-defined. The approach developed here aims at taking advantage of the user's capacities and feedback by allowing him to control parameters in real-time and to see the resulting visualization. In order to have fast transitions between visualizations resulting from different values of the parameter, interpolation on a grid is used as an approximation. The accuracy of this approximation is estimated using Procrustes analysis and can be adjusted through a threshold. Simulations provide an interpretation of this threshold and are validated on a real dataset.

Categories and Subject Descriptors (according to ACM CCS): [Human-centered computing]: Information visualization—

1. Introduction

With the increasing possibilities to have access to data, dealing with and analyzing very large datasets is a usual task today. The datasets can be large in the number of observations as well as in the number of dimensions. Visualization is a powerful tool in the first steps of data analysis: it gives a fast, intuitive and comprehensive view of the data. Many dimensionality reduction tools exist to visualize high-dimensional data. However, most algorithms implement a compromise between conflicting objectives. For example, it is impossible to project on a 2-dimensional space a 3-dimensional sphere, both without flattening and without tearing the sphere. Modern methods such as NeRV [VPN*10] and JSE [Lee12] explicit this compromise by a trade-off parameter to be tuned.

Two main possibilities exist to tune this parameter: the first one consists in defining an objective function, and to let the algorithm minimize this mathematical criterion. The second one is by trial and error, adjusting the parameter depending on the visualization obtained.

The first approach is the best one if a specific objective function to optimize exists. However it requires to know a priori what kind of visualization is wanted, which is contradictory to the idea that visualization is mostly used to get a first insight into the data. At that step it is hardly possible to express the goals into a precise mathematical form, which makes automatic setting of parameters difficult too.

The second approach has the advantage to introduce an interaction with the user. It allows us to take advantage of human intuition and background knowledge hard to write in mathematical terms [KKEM10] [STMT12]. However an important limitation in this approach is the time needed for computing each visualization. When dealing with visualization methods that use complex nonlinear optimization algorithms, the computation time might be incompatible with the real-time requirements of human interaction. Reduced computation time can be obtained by use of adapted algorithms [IMO09] [IM12].

This paper develops the second approach using approximations to restrict the computation time. The main challenge is to balance accuracy of approximations with computation time. Section 2 gives an illustration of the problem; Section 3 details the proposed method while Section 4 emphasizes on the choice of parameter values.

2. Example

Many nonlinear dimension reduction techniques use a parameter to tune a trade-off between two conflicting objectives. In the NeRV [VPN*10] and JSE [Lee12] methods the parameter is the relative weight given to false positives and false negatives. The parameter to tune can also be a trade-off between different types of features, as in MRE [MH05], or the rate of supervision as in [AG11] or [dRKO*03]. The choice of this parameter can have a large influence on the final result as can be seen in Figures 1 and 2. Figure 1 shows different projections, going from crushing to peeling a sphere. On Figure 2, the clusters (identified by colors) are more or less clearly separated depending on the balance between two types of features. The brown cluster is very clear for $\lambda = 1$, and all black points are concentrated for $\lambda = 0.4$. Separation between the red and blue clusters at $\lambda = 1$ can be improved by using $\lambda = 0.4$. No visualization is better a priori since those are just different views of a single problem.

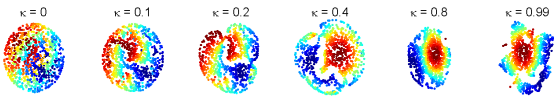

Figure 1: *NeRV projection of a 3D sphere depending on κ, trade-off between false positives and false negatives.*

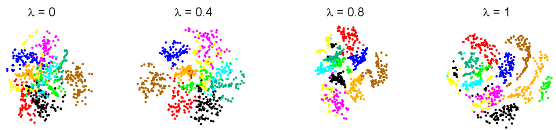

Figure 2: *JSE projection of [AA97] depending on λ, trade-off between two types of features.*

3. Proposed method

In order to allow the user to efficiently explore the configurations resulting from different values of the parameters, it is hardly possible to rely on real-time computations. This would require a computation time (of a new visualization) in less than e.g. 1 second, which is not compatible with most efficient nonlinear dimension reduction tools (on a stand-alone computer). The solution suggested here is to pre-compute a n-dimensional grid of projections (where n is the number of parameters) and to interpolate for other parameters values. Of course the key question is the resolution of the grid, which directly influences the computation time.

To avoid interpolating between too different projections, dissimilarities between projections can be estimated using Procrustes analysis [SC70]. Depending on the accuracy wanted, a threshold value can be chosen below which configurations are considered as close enough to use the interpolation, otherwise a new projection is computed. This approach requires a few assumptions: the method used to visualize data must be known and accessible, and the point to point correspondence between all visualizations is known ; these hypotheses are commonly met. Another underlying assumption is that the projection algorithm gives sufficiently continuous and smooth visualizations with respect to a change in the parameters. In algorithms with local minima some precautions could be needed, as fixing random components [GFVLD13].

This approach leads to two main questions:

- How to align the different visualizations in order to efficiently match them? Procrustes analysis aims to align at best (in the least squares sense) two sets of points using rigid transformations.
- How to avoid interpolation between too different projections, i.e. how to define the grid accurately enough? The distance minimized by Procrustes analysis may be used to evaluate the similarity between two visualizations.

3.1. Visualizations alignment using Procrustes analysis

To facilitate the transitions between different visualizations, the pre-computed projections need to be aligned as best as possible. Knowing the correspondence between points of each projection allows us to use Procrustes analysis [SC70] that centers, rescales and rotates the set of points to match it as best as possible with a reference (in the least square sense). To align the set $A \in R^{p\times q}$ on the set $B \in R^{p\times q}$, we are looking for a transformation $\tilde{B} = cAT + \mathbf{1}g$ where c is a scalar scaling coefficient, T is an $q\times q$ orthogonal rotation matrix , g is a $1\times q$ translation vector and $\mathbf{1}$ is a $p\times 1$ vector of ones, such that

$$B^* = \arg\min ||(B-\tilde{B})||_F^2 = \arg\min tr((B-\tilde{B})'(B-\tilde{B})) \quad (1)$$

where F stands for Frobenius norm. Let $Q = I - \mathbf{1}\mathbf{1}'/p$ be a centering matrix, and VDW' a singular value decomposition of $A'QB$. Then solutions to 1 are [SC70]

$$\begin{aligned} T &= VW' \\ c &= tr(T'A'QB)/tr(A'QA) \\ g &= (B-cAT)'\mathbf{1}/p. \end{aligned}$$

Therefore at the optimum the criterion to minimize is equal to $||(B-B^*)||_F^2 = trB'QB - (trT'A'QB)^2/trA'QA$.

However this criterion is neither symmetric nor scale-independent, two properties very desirable to compare projections. One way to overcome those limitations is to divide $||(B-B^*)||_F^2$ by $trB'QB$ [LJC74]. The final criterion, hereafter refered to as Procrustes value, is then

$$pv(A,B) = 1 - (trT'A'QB)^2/tr(A'QA)tr(B'QB) \quad (2)$$

which is symmetric and [0 1] bounded. Value of 0 is naturally reached when $A = B$. This last criterion allows us to measure in a general way if matrices A and B match better than matrices C and D.

The alignment of the visualizations computed with the grid values of the parameters can be done at once before any interpolation, or online. Pre-alignment on a grid of $n > 1$ parameters is less straightforward and depends on the path chosen from the reference configuration, since Procrustes analysis allows to align two sets of points at once only. To deal with more than two configurations at once, a possibility is to use the generalized Procrustes analysis [Gow75].

3.2. Grid precision

Once the grid-based visualizations are aligned, the next step is to evaluate if the grid is accurate enough. The assumption is that if two configurations A_{λ_1} and B_{λ_3} (corresponding to parameter values λ_1 and λ_3 respectively) are similar enough, then the transition between the visualizations will be smooth enough to use interpolated X'_{λ_2} as a good approximation of the real visualization X_{λ_2} (corresponding to parameter value λ_2). More mathematically, for $\lambda_1 < \lambda_2 < \lambda_3$ and if $similarity(A_{\lambda_1}, B_{\lambda_3})$ is small enough then we assume $similarity(X_{\lambda_2}, X'_{\lambda_2}) < similarity(A_{\lambda_1}, B_{\lambda_3})$ (see Figure 3).

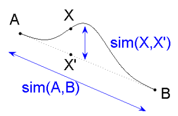

Figure 3: *Illustration of the assumption of smoothness*

The similarity between configurations can be evaluated by the pv value provided by Procrustes analysis in Equation 2, noted pv_{grid} for two successive configurations on the grid. In case of a $n > 1$ dimensional grid, the maximum value between pairs of configurations on the square ($n = 2$) or (hyper-)cube ($n > 2$) is used (bi-, tri-,...,interpolation is used). The idea is then to determine a threshold t such that if $pv_{grid} < t$ interpolation is considered to be good enough, otherwise new exact projections are computed to refine the grid in this area of the parameter space.

4. Procrustes value interpretation

The main difficulty with the proposed procedure is to define what a 'good enough' approximation means, and to choose a suitable threshold value t accordingly. It implies to interpret pv in terms of the difference between configurations, or more precisely in terms of point displacements between the two examined configurations. A bigger pv implies a weaker match between the matrices compared, but the intuitive interpretation of values between 0 and 1 is not straightforward. This section aims at evaluating the influence of the point configuration and of the type of perturbations on pv by means of simulations; then simulations results are confronted to results on a real dataset.

4.1. Understanding Procrustes parameter

To understand the relationship between the pv value and the point displacements, simulations were performed where random distributions of points are deliberately perturbed. As the main aspect in Procrustes analysis is the magnitude of the displacements, the orientation of the perturbations is drawn at random from a uniform variable in simulations.

How to quantify the perturbations? In order to study the relation between the Procrustes value and the amount of perturbations applied to the points, we first have to define how to quantify the latter. As scale effects and the number of points (for a given distribution) should not intervene, the perturbation magnitude $|pert|$ must be proportional to the distance to the neighbors: $|pert_i| \sim f(d_i(k))$, where i is the perturbed point , f is a probability distribution function, and $d_i(k)$ the distance from i to its k nearest neighbor. As exchanging positions of two neighbor points in a 10 points dataset has much more impact than in a 100 points dataset, the neighborhood must be proportional to the total number of points. So we take $k = l * p$ with p the number of points and l in $[0\ 1]$ represents the mean perturbation in terms of neighborhood proportion. Taking $l = 0.1$ means that in average x_i will be displaced of the distance necessary to cover its 10% nearest neighbors. The distribution f is chosen here as an half normal distribution in order to have a lot of small perturbations and a few large ones. Choosing $f = |N(0, d_i(k)\sqrt{2\pi}/2)|$ gives a mean displacement $d_i(k)$.

Influence of initial configuration. Figure 4 shows the different types of distributions perturbated; the respective distributions are uniform, Gaussian, composed of three separate clusters, of one cluster with 3 outliers, and of softly separated clusters.

Figure 5 shows the Procrustes value with respect to parameter l, for the five distributions. As expected the Procrustes value increases when the neighborhood grows. Two curves have a clearly different behavior which can be easily explained. For the three clearly separated clusters, the two large steps in the Procrustes value appear when the neighborhood grows to cover a second cluster. The same phenomenon appears much earlier with the dataset with 3 outliers, because the smaller clusters contain only one observation.

Figure 4: *Example of initial configurations to perturbate*

Considering the closeness of most of the curves in Figure 5 and the fact that the case with the perfectly separable clusters is the less natural one, the configuration with the Gaussian distribution is chosen as reference in the following.

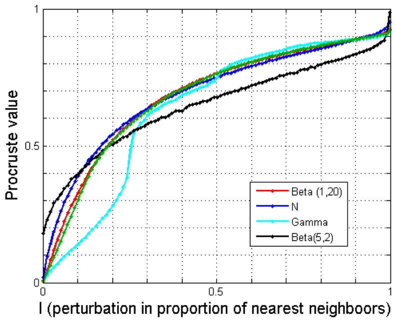

Figure 5: *Mean Procrustes values with respect to the mean displacement in terms of % of the nearest neighbors, for different initial configurations.*

Influence of the perturbation distribution. To see influence of the perturbation distribution, the chosen (Gaussian) distribution of points is perturbed by the $Beta(1,20)$, N, $Gamma(2.9,1/2.9)$ and $Beta(5,2)$ distributions (the order of this list corresponds to increasing modes of the perturbation distribution). Figure 6 shows the evolution of the Procrustes values for the different types of perturbation distributions. The difference between the curves is easily explained: as Procrustes analysis is l_2-norm based, the distributions with a concentration of the perturbations (or mode) near zero, and so implying fewer but larger strong perturbations, tend to give larger Procrustes values.

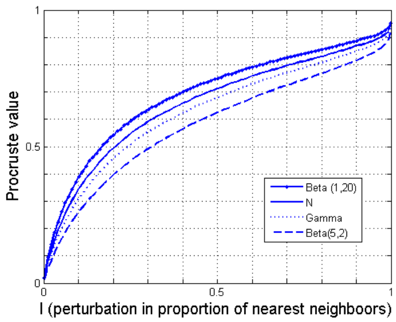

Figure 6: *Mean Procrustes values with respect to the mean displacement in term of % of the nearest neighbors, for different perturbation distributions.*

4.2. Test on real data

To confront simulations to results on a real dataset, interpolated visualizations based on a rough grid were compared to the exact visualizations. The latter correspond here to the result obtained using the dimensionality reduction algorithm from [Lee12], used as an example. The parameter to adjust can be interpreted as a way to choose the trade-off between false positives and false negatives. A second parameter was added to allow a balance between two types of features. Exact visualizations on a 2-parameters grid with values of $[0,1] \times [0,1]$ by step of 0.025 were compared to interpolated visualizations based on a grid with step of 0.2. The dataset used is the handwritten digits *pendigits* dataset [AA97] available on [FA10]. The first type of features is made of the images themselves (16×16 pixels), the second one consists of eight successive pen points on a two-dimensional coordinate system (8×2 coordinates).

Procrustes value depending on neighborhood perturbations. Figure 7 (left) shows the same type of results for the *pendigits* dataset as in Figure 6. The shape of the curve is very similar to the simulated ones, however the Procrustes values increase more slowly with the perturbations: simulations seem to slightly overestimate real Procrustes value.

Estimated Procrustes value depending on neighborhood perturbations. Comparing pv_{grid} (cf. Section 3) to l (cf. Section 4.1) in Figure 7 (right), a threshold t of 0.5 (corresponding to $l < 0.2$) seems a good trade-off between accuracy and the number of recomputed visualizations.

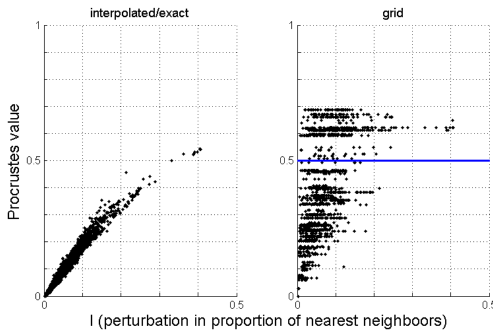

Figure 7: *With respect to l, Procrustes values between: -left) interpolated and exact visualizations (A and B in Figure 3) -right) grid projections used to interpolate (X and X′ in Figure 3).*

5. Conclusion

Many dimensionality reduction methods imply to tune a parameter corresponding to a trade-off between conflicting objectives. The proposed approach is to let the user choose between different parameter values by means of visualization. To avoid heavy recomputations for each parameter value, approximations are provided using linear interpolation between precomputed configurations. Accuracy of the approximations is evaluated using Procrustes analysis. The compromise between accuracy and computation time can be adjusted via a threshold value t. An interpretation of the value of t in terms of points displacements is given to help the user choose an appropriate value. The main hypothesis behind this work is the stability of the projections, which remains an open question in the field of dimension reduction.

References

[AA97] ALIMOGLU F., ALPAYDIN E.: Combining multiple representations and classifiers for pen-based handwritten digit recognition. In *Proceedings of the 4th International Conference on Document Analysis and Recognition* (1997), pp. 637–640. 2, 4

[AG11] ALIPANAHI B., GHODSI A.: Guided locally linear embedding. *Pattern Recogn. Lett. 32*, 7 (2011), 1029–1035. 2

[dRKO*03] DE RIDDER D., KOUROPTEVA O., OKUN O., PIETIKÄINEN M., DUIN R. P. W.: Supervised locally linear embedding. In *Proceedings of the 2003 joint International Conference on Artificial Neural Networks and Neural Information Processing* (2003), pp. 333–341. 2

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

[GFVLD13] GARCIA-FERNANDEZ F. J., VERLEYSEN M., LEE J. A., DIAZ I.: Sensitivity to parameter and data variations in dimensionality reduction techniques. In *Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2013), pp. 95–100. 2

[Gow75] GOWER J. C.: Generalized procrustes analysis. *Psychometrica 40*, 1 (march 1975), 423–427. 3

[IM12] INGRAM S., MUNZNER T.: Glint: An mds framework for costly distance functions. In *SIGRAD* (2012). 1

[IMO09] INGRAM S., MUNZNER T., OLANO M.: Glimmer: Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics 15*, 2 (2009), 249–261. 1

[KKEM10] KEIM D. A., KOHLHAMMER J., ELLIS G., MANSMANN F.: *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010. 1

[Lee12] LEE J. A.: Type 1 and 2 mixtures of divergences for stochastic neighbor embedding. In *Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2012), pp. 525–530. 1, 2, 4

[LJC74] LINGOES J. C. S. P. H.: Alternative measures of fit for the schonemann-carroll matrix fitting algorithm. *Psychometrica 39*, 4 (1974), 423–427. 2

[MH05] MEMISEVIC R., HINTON G.: Multiple relational embedding. In *Advances in Neural Information Processing Systems 17*. 2005, pp. 913–920. 2

[SC70] SCHONEMANN P. H., CARROLL R. M.: Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika 35*, 2 (1970), 245–255. 2

[STMT12] SEDLMAIR M., TATU A., MUNZNER T., TORY M.: A taxonomy of visual cluster separation factors. *Computer Graphics Forum (Proc. EuroVis) 31*, 3 (2012), 1335–1344. 1

[VPN*10] VENNA J., PELTONEN J., NYBO K., AIDOS H., KASKI S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res. 11* (2010), 451–490. 1, 2