On the visualization of high-dimensional data

Pierrick Bruneau

On the visualization of high-dimensional data

Pierrick Bruneau *

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

Abstract

Computing distances in high-dimensional spaces is deemed with the *empty space phenomenon*, which may harm distance-based algorithms for data visualization. We focus on transforming high-dimensional numeric data for their visualization using the kernel PCA 2D projection. Gaussian and p-Gaussian kernels are often advocated when confronted to such data; we propose to give some insight of their properties and behaviour in the context of a 2D projection for visualization. An alternative approach, that directly impacts the distribution of distances, is proposed. It also allows the indirect control of the distribution of the eventual kernel values as generated by the Gaussian kernel function. Finally, such projections induce some artifacts, which, if not handled, should not be ignored.

1 Distribution of distances in high-dimensional spaces

In high-dimensional spaces, normalized pairwise Euclidian distances tend to become all equal to 1 (see [3, section 1.4] for a justification). This is a corollary of the well-known *curse of dimensionality*, or *empty space phenomenon*. To illustrate this, we consider an artificial dataset of 3000 elements and 500 dimensions, each value being drawn independently from a uniform law in [0,1]. The dataset thus lies in the 500-dimensional unit hypercube. The histogram of pairwise distances between elements in the dataset (figure 1) clearly illustrates the claim of their being excessively biased towards 1.

This means that distance-based visulization methods (e.g. graph embedding that would use distances to discover a topology) would complicate the interpretation of the data by a user, all elements being equally dissimilar.

*bruneau@lippmann.lu

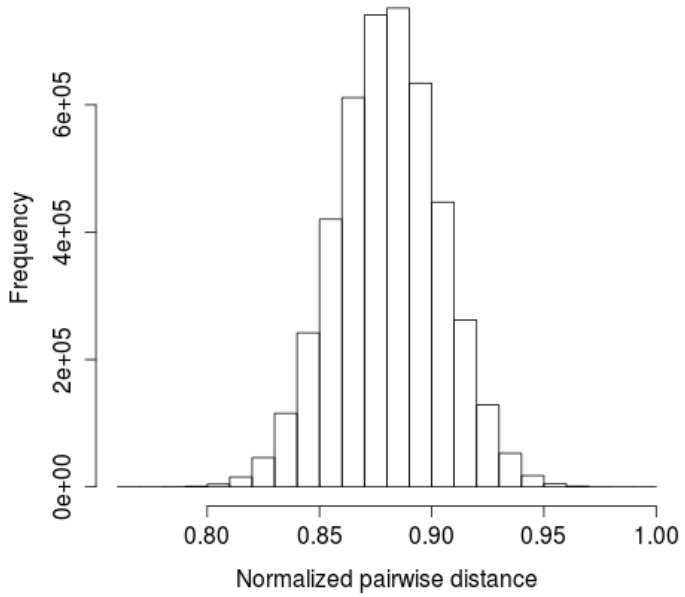

Figure 1: Distribution of pairwise Euclidian distances in the artificial dataset.

2 Kernel PCA projection for visualizing high dimensional data

The kernel PCA is the *kernelized* version of the PCA, a popular projection method. It operates on a kernel matrix (i.e. positive semi-definite similarity matrix), and extracts *non-linear principal manifolds* underlying the similarity matrix (see appendix A for details). The method maps these manifolds on a vector space: thus, we can build approximate, non-linear, 2D projections of high-dimensional data, by selecting the 2 dominant eigen-dimensions, and the values taken by the data elements on these.

Our intuition is that, with an adequate kernel function and matrix, this projection will lead to meaningful representations of a data set, from the distance distribution point of view.

3 Choice of a kernel function

As stated in the previous section, we aim at finding a suitable kernel for high-dimensional numeric data, i.e. that is little sensitive to the curse of dimensionality. A properly parametrized gaussian kernel function was successfully used in such situations [5]:

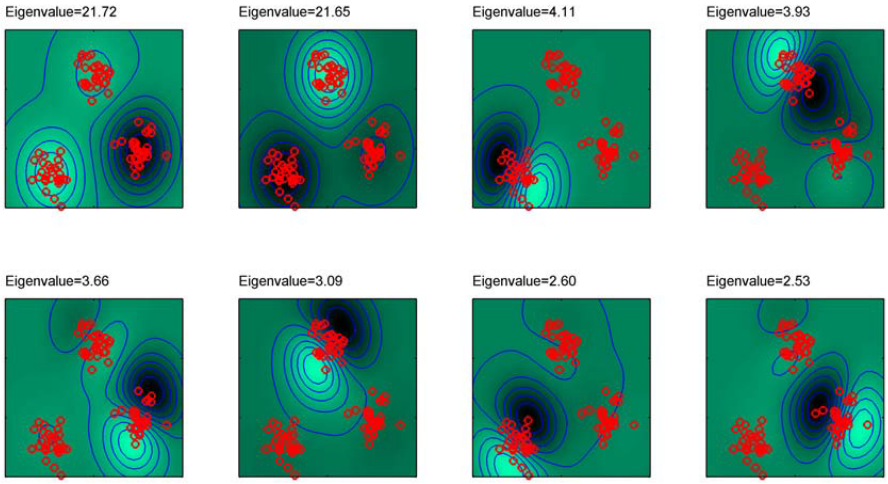

Figure 2: Example of kernel PCA major eigen-dimensions, with a Gaussian kernel applied to a synthetic data set in two dimensions. The contour lines and colour luminance indicate how values in the original data space are mapped in the eigen-dimensions (quoted from [2])

.

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d_{L2}(\mathbf{x}, \mathbf{x}')^2}{\sigma^2}\right), \tag{1}$$

Where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, $d_{L2}(.,.)$ denotes the Euclidian distance, and $\sigma = \sup d_{L2}(.,.)$. If $\mathbf{x}$ and $\mathbf{x}'$ are members of a data set $\mathbf{X}$, the bound may empirically be set with $\max_{\mathbf{x},\mathbf{x}' \in \mathbf{X}} d_{L2}(\mathbf{x}, \mathbf{x}')$. This kernel function can be interpreted as a smoothed neighborhood detector. As can be seen in figure 2, values in the mapped vector space indicate the closeness to some local dense pattern in the original data.

Using the kernel function (1), and the kernel PCA projection method, the artificial dataset of the previous section was mapped to a 2 dimensional projection. It is shown on figure 3a, and the associated histogram of pairwise distances, taken in the transformed 2D space, is given in figure 4a. It exhibits a much wider distribution, which emphasizes the interest of the method in order to represent the data.

The Gaussian kernel has a single drawback: the distribution of its values is dimensionality dependent. For our 500-dimensional example, it is given in figure 5a. In fact, even if we are able to perform the kernel PCA projection for this example, the Gaussian kernel is actually not completely insensitive to the curse of dimensionality : the higher the dimensionality, the sharper it is peaked just above its lower bound, $\exp(-1)$. In extreme cases, this might lead to numeric issues, such as unstable eigen-decompositions.

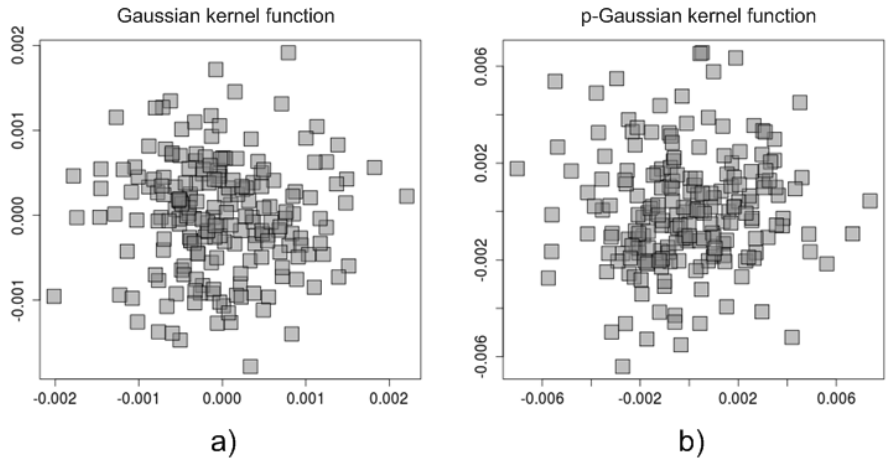

Figure 3: 2D projections of the artificial dataset with the kernel PCA method, using either the Gaussian (a) or the p-Gaussian (b) kernel function. For the sake of clarity, a subset of 200 elements, extracted from the collection of 3000 elements, is solely displayed.

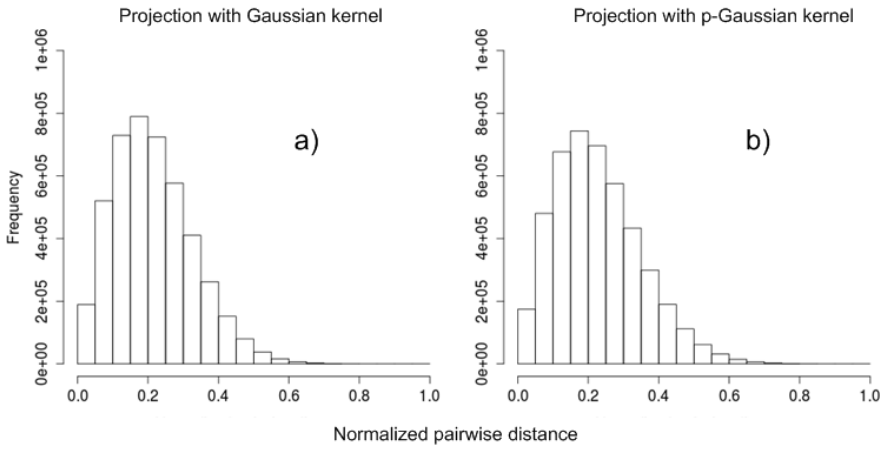

Figure 4: Distribution of pairwise Euclidian distances in the artificial dataset when represented in the kernel PCA transformed-2D space, using either the Gaussian (a) or p-Gaussian (b) kernel function.

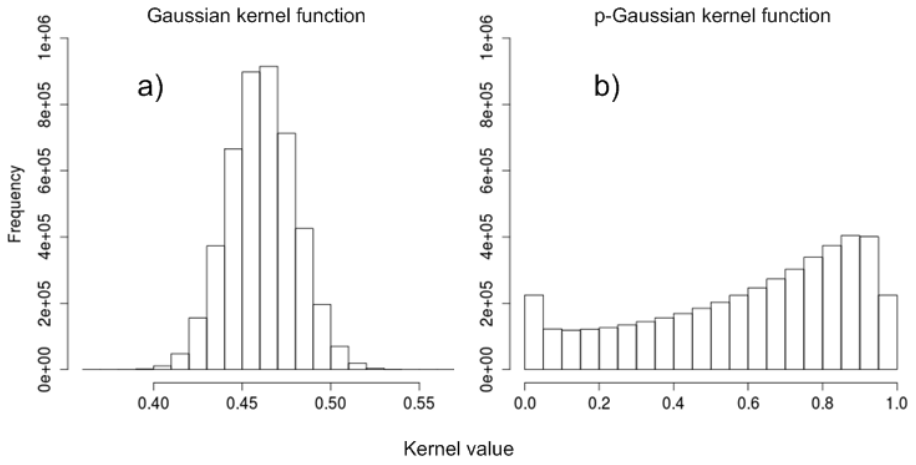

Figure 5: Distribution of values returned by the Gaussian (a) and p-Gaussian (b) functions for our artificial dataset.

This is what motivated the p-Gaussian kernel function, a variant that explicitly takes account of the space dimensionality and original distance distribution [4]:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d_{L2}(\mathbf{x}, \mathbf{x}'))^p}{\sigma^p}\right), \tag{2}$$

This function was adjoined by empirical formulas for setting p and σ, designed to ensure that the kernel values match the cumulative distribution of the distances in the original space, irrespective of its dimensionality:

$$p = \frac{\ln(\frac{\ln 0.05}{\ln 0.95})}{\ln \frac{d_{L2}^{95\%}}{d_{L2}^{5\%}}}, \quad \sigma = \frac{d_{L2}^{95\%}}{(-\ln 0.05)^{\frac{1}{p}}} = \frac{d_{L2}^{5\%}}{(-\ln 0.95)^{\frac{1}{p}}}, \tag{3}$$

with $d_{L2}^{5\%}$ (resp. $d_{L2}^{95\%}$) the 5% (resp. 95%) percentile of the cumulative distribution of d_{L2} [1]. This latter kernel function was also used to apply the kernel PCA 2D projection to our synthetic 500-dimensional dataset. This led to the projection in figure 3b, with the associated distance distribution shown in figure 4b. We see that, up to a scale factor, it does not significantly differ from the classical Gaussian kernel with this respect.

[1] In the referenced paper, $d_{L2}^{5\%}$ and $d_{L2}^{95\%}$ have been mistakenly swaped in the expressions for σ. A corrected version is reported here.

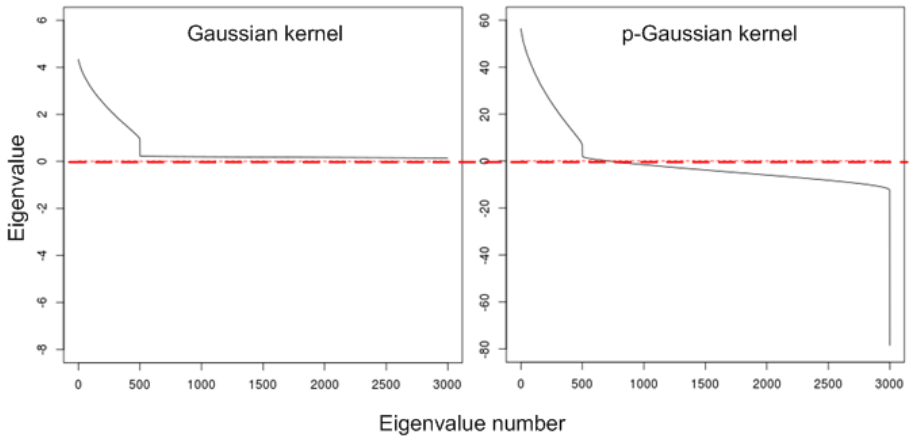

Figure 6: Eigenvalues, in decreasing order, for the Gaussian and p-Gaussian kernel evaluated on our artificial 500-dimensional dataset. For legibility, the first eigenvalue, 100 times higher in magnitude than any other eigenvalue in both cases, was omitted.

Examples of eigenvalue profiles are given in figure 6 for the Gaussian and p-Gaussian kernels. We first notice that, apart from the scale factor mentionned above, the profile of the 20% leading eigenvalues of both kernels is very similar. However, the remaining eigenvalues are negative for the p-Gaussian kernel, which states its non-positive semi-definiteness.

The kernel PCA 2D projection only requires the 2 major eigenpairs, which, provided the data is sufficiently plentiful, and non-degenerate, will always be associated to positive eigenvalues[1]. This is reflected by the similar projections, and associated distance profiles, given above.

In figure 5b, we see that the p-Gaussian kernel values are almost evenly distributed in $[0, 1]$. Then, practitionners may choose according to their targetted application: if the positive semi-definiteness is mandatory, the Gaussian kernel function is the only acceptable choice. If numerical stability is the most important, the p-Gaussian seems preferable.

4 Alternative to the p-Gaussian

The p-Gaussian implements the adaption to high dimensionalities with a specially crafted kernel function, that accounts for quantiles of the distribution of distances. In this section we explore an alternative solution.

[1]The Perron-Frobenius theorem only guarantees one real positive eigenvalue, and the matrix being real symmetric, we know all eigenvalues are real. But no further level of proof seems possible currently.

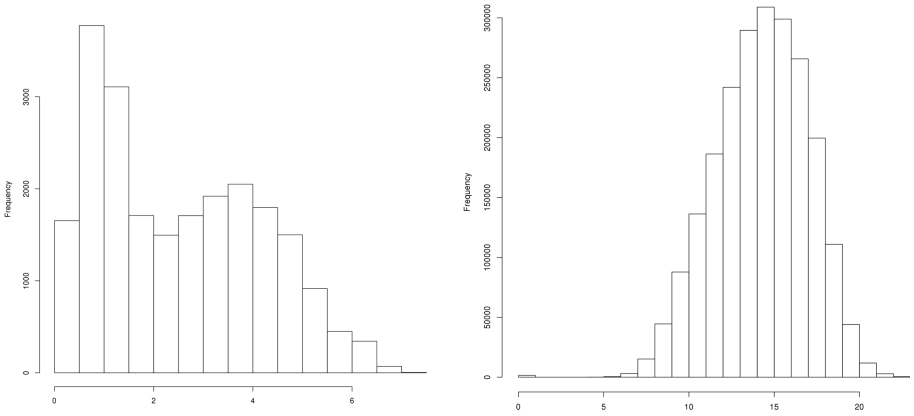

Figure 7: Distribution of pairwise distances for iris (left) and isolet (right).

Our derivations will be supported by graphics based on the *iris* data set (150 elements, 4 dimensions, and 3 classes with 50 elements per class) and the *isolet* vowels (1500 elements, 617 dimensions, and 5 classes with 300 elements per class).

As illustrated in Figure 1, and discussed earlier, pairwise high-dimensional distances tends to have a peaked distribution, reminiscent of the Gaussian. Depending on the extent of the dimensionality, it tends to be more or less sharply peaked. Figure 7 gives the distribution of pairwise distances for isolet, that shows the expected distribution. The latter figure also gives the distribution for iris, to illustrate that the distribution tends to be much more uniform for low-dimensional data.

Our goal is to somehow control the distribution of distances, and subsequent Gaussian kernel values, so that the eigendecomposition is stable. A possible solution would be to derive a monotonical uniform map of the pairwise distances to $[0, 1]$. Subsequent kernel values would then be distributed broadly in $[\exp(-1), 1]$ (instead of the narrow distribution such as in Figure 5a), independently of the underlying dimensionality.

If values are distributed according to a Gaussian, then applying the *cumulative distribution function*(cdf) of the Gaussian to these values is expected to result in a uniform distribution.

As the actual cdf is rather tedious to compute, we approximate the cdf of $\mathcal{N}(0, 1)$ with a sigmoidal function (see Figure 8). The distance values then just require standardization prior to applying the sigmoidal function. The resulting remapped distance distribution is shown for isolet in Figure 9, and is indeed much closer to uniformity.

Unfortunately, values obtained under this transformation may rather be coined as *dissimilarities*, as they are not valid distances any more (i.e. the triangular inequality is not guaranteed then). Thus, using this method raises

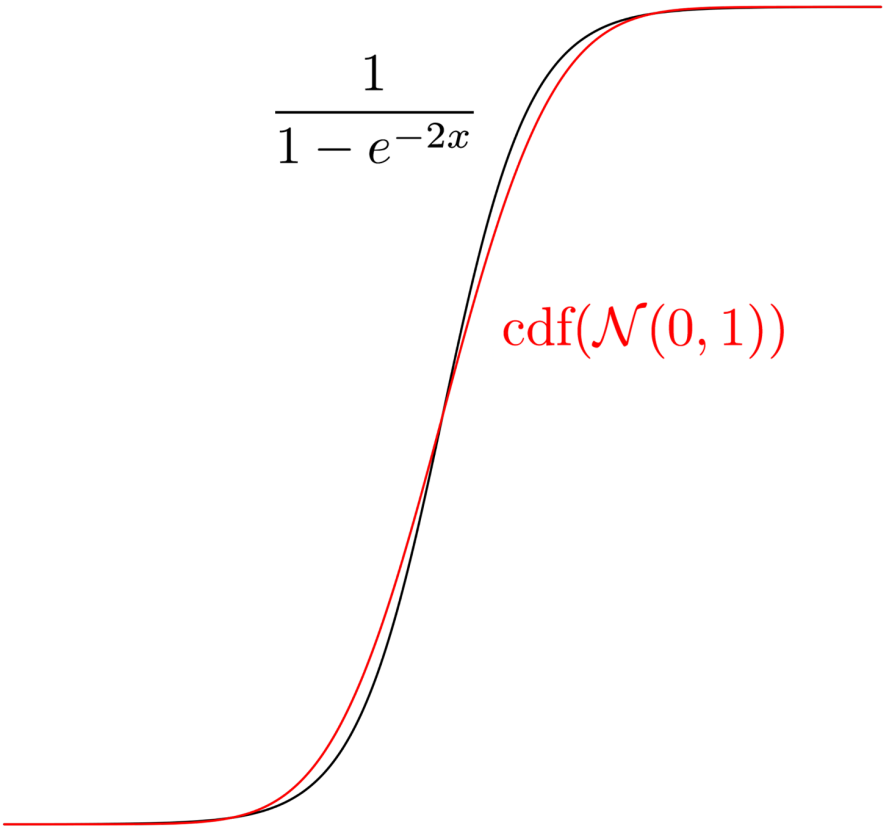

Figure 8: Illustration of the fit of the sigmoid function to the actual Gaussian cdf function.

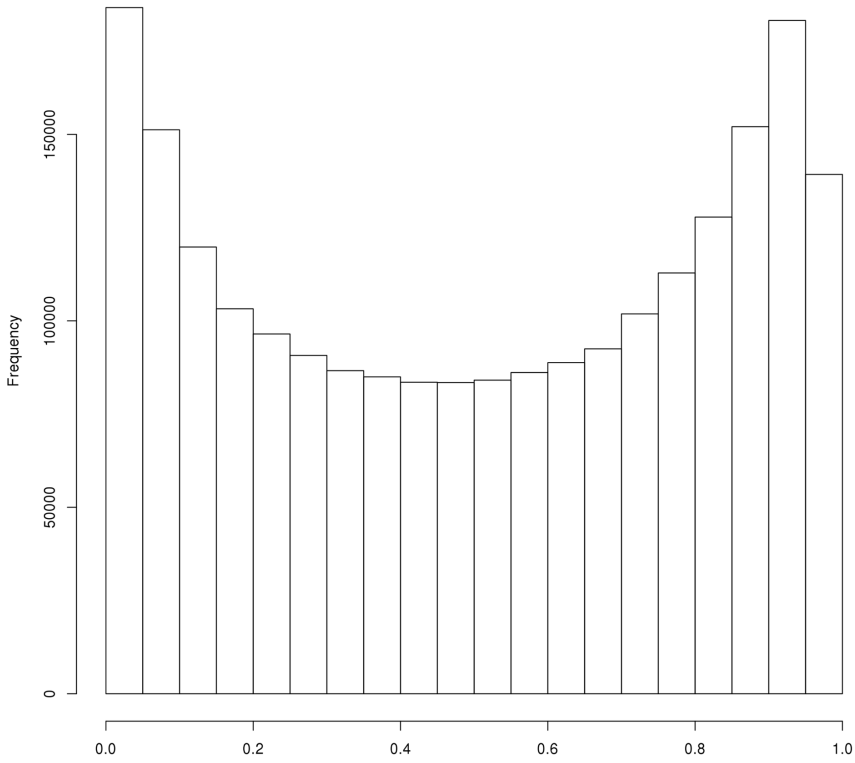

Figure 9: Distribution of pairwise dissimilarities in isolet, after transformation.

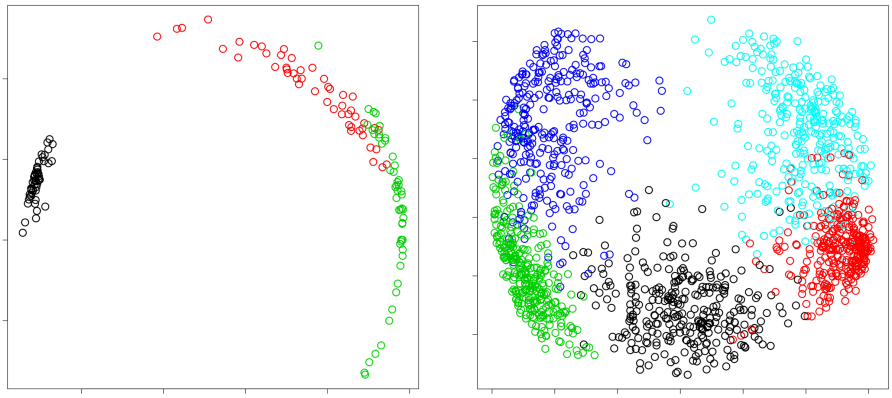

Figure 10: Projection resulting from the Gaussian kernel applied to remapped dissimialrities, for iris (left) and isolet (right).

the same theoretical issues as the p-Gaussian function (i.e. non positive semi-definiteness). Evaluations still have to be carried out to assess the influence of all these transformations in terms of a stress function.

The projection resulting from the application of the usual Gaussian kernel to the transformed dissimilarities is shown in Figure 10. For isolet, the result seems satisfactory (even if to be confirmed in terms of stress). However for iris the result illustrates some kind of degeneracy. As stated earlier, the transformation assumes a Gaussian distribution of the distance values, and Figure 7 gives a hint of the inadequacy for iris. Programmatically, Shapiro-Wilk tests may be employed to assess if the gaussianity may be rejected safely (if not, our approach must be taken). Let us note that with very large samples, statistical tests tend to be too powerful: as the distribution is only roughly Gaussian, rejection is almost certain in the limit of large samples. To balance the power of the test, one may consider using only a subsample from the set of distance values, yet sufficiently large to reject when distances are clearly not Gaussian. In our case, using samples of 100 elements was rather effective (almost always $p < 10^{-2}$ for iris, and around 0.3 for isolet). As this scheme relies on random draws, multiple trials are necessary for robust results.

5 Stress properties of a kernel PCA projection

When projecting d-dimensional data to a 2-dimensional space ($d > 2$), there is necessarily some projection artifacts, i.e. some distortion induced by the transformed 2D space with respect to the distribution of pairwise distances (see [1] for details on this matter).

In figure 11, we show the average compression and stretching artifacts pro-

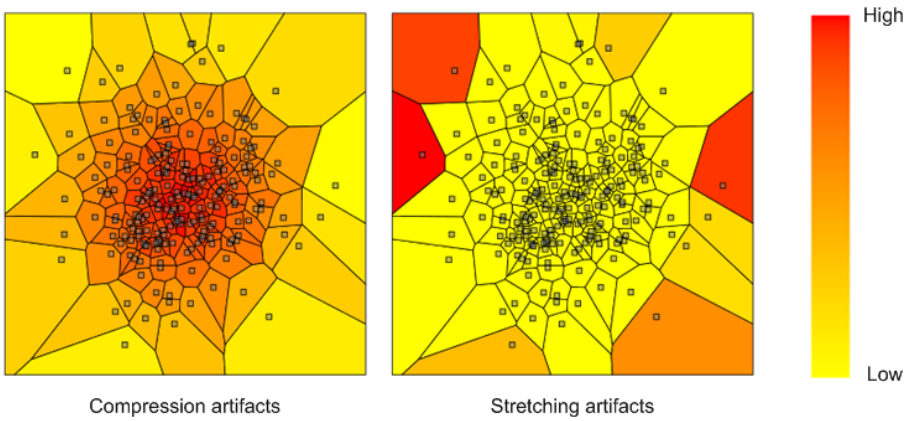

Figure 11: Compressing and stretching artifacts, represented for a subset of 200 randomly chosen elements. The distortion values are represented on a heat map scale, and their average for each element is mapped on its Voronoi cell.

duced by the application of the kernel PCA 2D projection to our dataset with the p-Gaussian kernel function. In brief, each element is matched with its tendency to have lower (compression) or higher (stretching) pairwise distances to other elements, in comparison to the distances in the original data space.

Figure 11 illustrates how the 2D projection method compensates the peaked distribution of pairwise distances in the original space, by compressing the elements in the center of the projection, and stretching the elements close to the projection boundaries. The information loss implied by the process is somehow materialized by the obtained vaguely "Gaussian shaped" distribution, whereas our knowledge of the ground truth generating process would lead us to expect a uniform distribution.

6 Software implementations

The toy experiments, and graphics shown in this paper were implemented with R. The kernel PCA projections were performed with the *semisupKernelPCA* package. The graphics mostly relied on the *patchPlot* and *deldir* packages. *irlba* was used for the fast extraction of the two major eigenpairs.

References

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

[2] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[3] M. A. Carreira-Perpinan. A review of dimension reduction techniques. Technical report, University of Sheffield, 1997.

[4] 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.

[5] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1999.

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

A Kernel PCA theory

Let us consider a set of elements $\mathbf{X} = \{\mathbf{x}_i\}_{i \in 1 \dots N}$, with values in some domain $\mathcal{X}$ (referred to as *original space* hereafter), and a nonlinear, unknown yet, transformation ϕ that projects any element $\mathbf{x}_i$ onto a point $\phi(\mathbf{x}_i) \in \mathbb{R}^M$ (called *feature space* in the remainder).

Assuming $\sum_{i=1}^{N} \phi(\mathbf{x}_i) = \mathbf{0}$, the sample covariance matrix of the image of $\mathbf{X}$ in the feature space is given as:

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T,$$

with the associated eigenvector equation:

$$\mathbf{C}\mathbf{v}_m = \lambda_m \mathbf{v}_m, \ m = 1 \dots M.$$

Considering the kernel function $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$, and following works by [6] and [2], this eigenvector problem can be transformed to:

$$\mathbf{K}\mathbf{a}_m = \lambda_m N \mathbf{a}_m, \ m = 1 \dots M, \tag{4}$$

with $\mathbf{K}$ the $N \times N$ matrix such that $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{a}_m$ a vector in $\mathbb{R}^N$. Let us note that the mapping ϕ does generally not have to be explicitly defined: indeed, any positive semi-definite matrix $\mathbf{K}$ was proven to be the dot product in some feature space, may it be infinite dimensional [2]. Thus, practitioners preferably design kernel functions directly, only caring about the positive semi-definiteness of the induced kernel matrices.

After solving (4) for its eigenvectors and eigenvalues, a set of M projection functions can be defined as follows:

$$y_m(\mathbf{x}) = \sum_{i=1}^{N} a_{mi} k(\mathbf{x}, \mathbf{x}_i).$$

Assuming eigenvalues in decreasing order, the 2D projection that captures the maximal variance in the feature space is then built with y_1 and y_2. The assumption $\sum_{i=1}^{N} \phi(\mathbf{x}_i) = \mathbf{0}$ can be released with the following modified kernel expression [2]:

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N,$$

with $\mathbf{1}_N$ the $N \times N$ matrix in which every cell has the value $\frac{1}{N}$.