Helsinki University of Technology
Publications in Computer and Information Science

Report E5
September 2006

NONLINEAR DIMENSIONALITY REDUCTION AS INFORMATION RETRIEVAL

Jarkko Venna Samuel Kaski

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Distribution:
Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science
P.O. Box 5400
FI-02015 TKK, Finland
Tel. +358-9-451 3267
Fax +358-9-451 3277

This report is downloadable at
http://www.cis.hut.fi/Publications/

ISBN 951-22-8379-4
ISSN 1796-2803

Nonlinear dimensionality reduction as information retrieval

Jarkko Venna
Adaptive Informatics Research Centre
Helsinki University of Technology
P.O. Box 5400
FI-02015 HUT
FINLAND
jarkko.venna@tkk.fi

Samuel Kaski
Adaptive Informatics Research Centre
Helsinki University of Technology
P.O. Box 5400
FI-02015 HUT
FINLAND
samuel.kaski@tkk.fi

Abstract

Nonlinear dimensionality reduction has so far been treated either as a data representation problem or as a search for a lower-dimensional manifold embedded in the data space. Neither approach has been designed to optimize the visualization capability, although information visualization is perhaps the main application. We conceptualize visualization as an information retrieval problem; a projection is good if neighbors of data points can be retrieved well based on the visualized projected points. This makes it possible to rigorously quantify goodness in terms of precision and recall. A method is introduced to optimize retrieval quality; it turns out to be an extension of Stochastic Neighbor Embedding, one of the earlier nonlinear projection methods, which focuses only on recall.

1 Introduction

Early nonlinear projection methods introduced a representation for the data and optimized the representations to minimize representation error. Most can be interpreted as multidimensional scaling (MDS) methods [2] which minimize some measure of preservation of pairwise distances between data points.

More recently, there has been a lot of interest in methods that construct the projection by searching for data manifolds embedded in the original data space. Isomap [12] infers the manifold through local neighborhood relationships, and visualizes it by MDS; Locally Linear Embedding (LLE) [9] approximates the manifold locally by linear surfaces; Laplacian Eigenmap (LE) [1] and Hessian Eigenmap (HLLE) [4], are very similar but based on graph theory; Alignment of Local Models (ALM) [13] and other similar approaches first fit local models to the data and then search for a transformation that aligns them globally. Finally, there are more heuristically derived but surprisingly well-performing algorithms, such as the Curvilinear Components Analysis [3].

In visualization applications, a problem with all the methods listed above is that they have not been designed for visualization. The cost functions of data representation methods, which measure for instance preservation of pairwise distances, are only indirectly related to the goodness of the resulting visualization. Manifold search methods, on the other hand, have been designed to find the "true" manifold which may be higher than two-dimensional, which is the upper limit for visualization in practice. Hence, evaluating goodness of visualizations seems to require usability studies which would be laborious and slow.

In this paper we view the visualization from the user perspective, as an information retrieval problem. Assume that the task of the user is to understand the proximity relationships in the original high-dimensional data set, and the task of the visualization algorithm is to construct a display that helps

in this task. For a given data point, the user wants to know which other data points are its neighbors, and the visualization should reveal this for all data points, as well as possible. If this task description matches what the user is doing, our analysis gives rigorous grounds for constructing a visualization algorithm.

Any visualization algorithm will make two kinds of errors: Some neighbors will be missed (which reduces *recall*) and some non-neighbors will be visualized as neighbors (which reduces *precision*). In information retrieval it is traditional to evaluate systems based on curves of precision vs. recall, or optimize the system to minimize some combination of the two measures. Our suggestion is to do the same in visualization.

It turns out (details below) that one of the manifold extraction methods, Stochastic Neighbor Embedding (SNE) [5], can be interpreted to optimize a smoothed version of recall. In this paper we introduce a measure of precision to the algorithm, and optimize a parametrized compromise between the two. In the resulting Neighbor Retrieval Visualizer (NeRV) the compromise can be tuned according to the relative costs of the two criteria. It turns out that the method outperforms its alternatives on a wide range of values of the compromise parameter.

2 Stochastic Neighbor Embedding

The SNE algorithm [5] was originally motivated as a method for placing a set of objects into a low-dimensional space in a way that preserves neighbor identities. Such a projection does not try to preserve pairwise distances as such, as MDS does, but instead the *probabilities* of points being neighbors.

A probability distribution is defined in the input space, based on the pairwise distances, to describe how likely it is that the point i is a neighbor of point j. The same is done in the low-dimensional output or projection space. The algorithm then optimizes the configuration of points in the output space, such that the original distribution of neighborness is approximated as closely as possible in the output space. The natural measure of approximation error between distributions is the Kullback-Leibler (KL) divergence, which is averaged over all points.

More formally, the probability p_{ij} of the point i being a neighbor of point j in the input space is defined to be

$$p_{ij} = \frac{\exp\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)^2}{\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_k)^2}{\sigma_i^2}\right)}, \tag{1}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$. The width of the Gaussian, σ_i, is set either manually or by fixing the entropy of the distribution. Setting the entropy equal to $\log k$ sets the "effective number or neighbors" to k.

Similarly, the probability of the point i being a neighbor of point j in the output space is defined to be

$$q_{ij} = \frac{\exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_k\|^2}{\sigma_i^2}\right)}. \tag{2}$$

The SNE algorithm searches for the configuration of points $\mathbf{y}_i$ that minimizes the Kullback-Leibler divergence D between the probability distributions in the input and output spaces, averaged over all points. The cost function is

$$E_{\text{SNE}} = E_i[D(p_i, q_i)] \propto \sum_i D(p_i, q_i) = \sum_i \sum_{j\neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{3}$$

where E_i is the average over data samples i.

Relationship to recall. It turns out that SNE has an interpretation as an information retrieval algorithm; it optimizes a smoothed form of recall as we will show next. Assume that the user wants to retrieve neighbors of each data point, and do that with the help of the visualization (output space) only.

To show the connection we need to define neighborhoods as step functions. The user is studying r neighbors in the output space, and her goal is to find a large proportion of the k "true" neighbors, that is, neighbors in the input space.

Technically, we assume the k closest points to be neighbors with a high probability and the rest with a very low probability. Define

$$p_{ij} = \begin{cases} a \equiv \frac{1-\delta}{k}, \text{ if point } j \text{ is among the } k \text{ nearest neighbors of } i \text{ in the input space} \\ b \equiv \frac{\delta}{N-k-1}, \text{ otherwise} \end{cases}, \quad (4)$$

where N is the total number of data points, k is the size of the desired neighborhood and $0 < \delta < 0.5$ gives the non-neighbors a very small probability.

Similarly, we define the probability of j being a neighbor of i in the output space by

$$q_{ij} = \begin{cases} c \equiv \frac{1-\delta}{r}, \text{ if point } j \text{ is among the } r \text{ nearest neighbors of } i \text{ in the visualization} \\ d \equiv \frac{\delta}{N-r-1}, \text{ otherwise} \end{cases}, \quad (5)$$

where r is the neighborhood size in the output space.

Now each Kullback-Leibler divergence in the cost function can be divided into four parts,

$$\begin{aligned} D(p_i, q_i) = &\sum_{p_{ij}=a, q_{ij}=c} a \log \frac{a}{c} + \sum_{p_{ij}=a, q_{ij}=d} a \log \frac{a}{d} + \sum_{p_{ij}=b, q_{ij}=c} b \log \frac{b}{c} + \sum_{p_{ij}=b, q_{ij}=d} b \log \frac{b}{d} \\ &= C_{\text{TP}} N_{\text{TP}} + C_{\text{MISS}} N_{\text{MISS}} + C_{\text{FP}} N_{\text{FP}} + C_{\text{TN}} N_{\text{TN}}. \quad (6) \end{aligned}$$

Here N_{TP} is the number of true positives, that is, points where the probability of being a neighbor is high in both spaces. The number of misses, points not being chosen for retrieval because of a low probability in the output space although the probability in the input space is high, is N_{MISS}. The number of false positives is N_{FP}; high in the output space but low in the input space. Finally the number of true negatives (low in both spaces) is N_{TN}. The C are the constant coefficients; $C_{\text{TP}} = (1-\delta)/k \log(r/k)$, and the rest analogously.

It is straightforward to check that if δ is very small, then the coefficients for the misses and false positives dominate the cost E_{SNE}, and moreover

$$\begin{aligned} &D(p_i, q_i) \approx C_{\text{MISS}} N_{\text{MISS}} + C_{\text{FP}} N_{\text{FP}} = \\ &N_{MISS} \frac{1-\delta}{k} \left(\log \frac{(N-r-1)}{k} + \log \frac{(1-\delta)}{\delta} \right) + N_{FP} \frac{\delta}{N-k-1} \left(\log \frac{r}{N-k-1} - \log \frac{(1-\delta)}{\delta} \right) \\ &\approx \left(N_{MISS} \frac{1-\delta}{k} - N_{FP} \frac{\delta}{N-k-1} \right) \log \frac{(1-\delta)}{\delta} \\ &\approx N_{MISS} \frac{1-\delta}{k} \log \frac{(1-\delta)}{\delta} = \frac{N_{MISS}}{k} C, \quad (7) \end{aligned}$$

where C is a constant. This is the cost function SNE would try to minimize, and hence it would maximize recall which is defined as

$$\text{recall} = \frac{N_{TP}}{k} = 1 - \frac{N_{MISS}}{k}. \quad (8)$$

In summary, with a step function as a neighborhood distribution, the SNE would optimize average recall. This result is mainly theoretical, however, since optimization with such step functions would be very difficult in practice. Instead, SNE uses a Gaussian neighborhood function which can be interpreted as a smoothed step function. With the Gaussian the recall turns into a smoothed recall which takes takes into account the sizes of the errors as well as their number.

3 Neighbor Retrieval Visualizer

Understanding SNE in the information retrieval sense opens new avenues for improving it. SNE maximizes (smoothed) recall, and it is well known that maximizing recall typically leads to low precision, and vice versa. In other words, SNE only optimizes one end of the spectrum.

If we want to maximize precision, we can reverse the direction of the Kullback-Leibler divergence in (3). For step functions and for small δ, it is straightforward to show, analogously to the previous section, that

$$D(q_i|p_i) \approx \frac{N_{FP}}{r} C, \tag{9}$$

where N_{FP} is the number of false positives and r is the number of retrieved points. Minimizing this would correspond to maximizing precision defined as

$$\text{precision} = 1 - \frac{N_{FP}}{r}. \tag{10}$$

Hence, by reversing the direction of the Kullback-Leibler divergence in the cost function we get a method that focuses on gaining a high precision. Again, we could switch to Gaussian neighbor distributions instead of step functions, to get an algorithm that is analogous to SNE but that would maximize smoothed precision instead of smoothed recall.

In practice it would be best to optimize a compromise. If we can assign a relative cost λ to misses and $(1 - \lambda)$ to false positives, then the total cost function to be optimized should be

$$\begin{aligned} E_{\text{NeRV}} &= \lambda E_i[D(p_i, q_i)] + (1 - \lambda) E_i[D(q_i, p_i)] \\ &= \lambda \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} + (1 - \lambda) \sum_i \sum_{j \neq i} q_{ij} \log \frac{q_{ij}}{p_{ij}}. \end{aligned} \tag{11}$$

For step functions and small δ this would reduce to the total information retrieval cost, and for Gaussian functions as in SNE it can be interpreted as a smoothed cost. We call the new method that optimizes (11) *Neighbor Retrieval Visualizer (NeRV)*, since it interprets the visualization problem as a problem of retrieving neighbors based on the visualization.

By setting the parameter $\lambda \in [0, 1]$ we choose whether we want to focus more on the probabilities that are high in the input space (recall) or in the output space (precision). When $\lambda = 1$ the method becomes SNE and when $\lambda = 0$ it focuses purely on avoiding false positives.

We optimize the cost function using a conjugate gradient method. A heuristic but very effective way of avoiding local minima is to initialize the optimization by starting with a large width of the Gaussian, σ_i^2, and reducing it stepwise after each optimization step until the final value is reached. After this initialization, normal conjugate gradients are run with a fixed Gaussian.

4 Empirical Comparison

We compared the performance of NeRV to alternative methods on three data sets; the first is a small artificial set and the two others very high-dimensional real-world sets.

Thick S-curve. The first data set is a simple toy set sampled from a folded low-dimensional manifold, a two-dimensional S-shaped curve in a three-dimensional space. The 1000 data points were constructed as follows. First, the data was uniformly sampled from a two-dimensional S-shaped sheet. Then, to give the manifold a thickness, a spherical normally distributed displacement was added to each point.

Mouse gene expression. The second data set is a collection of gene expression profiles from different mouse tissues [11]. Expression of over 13,000 mouse genes had been measured in 45 tissues. We used an extremely simple filtering method, similar to that originally used in [11], to select the genes for visualization. Of the mouse genes clearly (average difference in Affymetrix chips, AD > 200) expressed in at least one of the 45 tissues (dimensions), a random sample of 1600 genes (points) was selected. After this the variance in each tissue was normalized to unity.

Gene expression compendium. The third data set is a large collection of human gene expression arrays [10]. (The normalized expression compendium is available from `http://dags.stanford.edu/cancer`.) Since the current implementations of all methods do not tolerate missing data we removed samples with missing values altogether. First we removed genes that were missing from more than 300 arrays. Then we removed the arrays for which values were still missing. This resulted in a data set containing 1278 points and 1339 dimensions.

Methods. The performance of NeRV was compared with the following dimensionality reduction methods: Principal Component Analysis (PCA) [6], metric MultiDimensional Scaling (MDS) [2], Locally Linear Embedding (LLE) [9], Laplacian Eigenmap (LE) [1], Hessian Eigenmap (HLLE) [4], Isomap [12], Alignment of Local Models (ALM) [13], Curvilinear Component Analysis (CCA) [3] and Curvilinear Distance Analysis (CDA) [8], which is a variant of CCA that uses graph distances to approximate the geodesic distances in the data.

Each method that has a parameter k for setting the number of nearest neighbors was tested with values of k ranging from 4 to 20, and the value producing the best results was selected. Methods that may have local optima were run 10 times with different random initializations and the best run was selected. For the NeRV algorithm we set the effective number of neighbors to 20 (without optimizing it further).

4.1 Results

We used three pairs of performance measures to compare the methods. The first one comes directly from the NeRV cost function: $E_i[D(p_i, q_i)]$ measures smoothed recall and $E_i[D(q_i, p_i)]$ smoothed precision. We plot the results all methods on the plane spanned by the two measures. NeRV forms a curve parametrized by λ (Fig. 1 top row). NeRV was clearly the best performing method on all three data sets, using this pair of measures.

Although the NeRV cost function is arguably a measure worth optimizing, we verified the results with two other sets of performance measures. Since our motivation comes from information retrieval we will plot standard precision–recall curves, as a function of the number of neighbors chosen from the output space. Finally we will use a pair of measures that is analogous to our smoothed precision and recall, namely the trustworthiness and continuity used in [7]. Trustworthiness measures how many of the neighbors defined in the output space are neighbors also in the input space, and continuity the other way around. Errors are penalized according to the rank distance from the neighborhood.

The precision–recall behavior of the methods is illustrated in the middle row of Figure 1. For computational reasons the curves are obtained by varying the number of retrieved neighbors, without re-computing the visualizations, although we know that for best results the effective number of neighbors in the width of the Gaussian smoother should match the number of retrieved neighbors. Hence the results will not be optimal.

In the precision-retrieval curves, usually when λ is small the precision becomes higher in the leftmost end of the curve (where the number of retrieved neighbors is small) than with a large λ. Analogously, a large λ seems to lead to better precision in the rightmost end. This is clear on the thick s-curve data set as well as on the mouse gene expression data set. It is unclear why the best precision is achieved with $\lambda = 1$ on the gene expression compendium. The CDA algorithm performed very well in terms of precision, being the best on all three data sets, followed by CCA and NeRV. NeRV with $\lambda = 1$ gained the best recall values on the two bioinformatics data sets but was superseded slightly by PCA on the thick s-curve data set.

The results of the trustworthiness and continuity measures (Fig.1, bottom row) are very similar to the ones from the Kullback-Leibler plots in the top row. One difference is that on the two bioinformatics data sets the highest trustworthiness was gained with a λ value that was in the middle of the scale. One explanation for this could be the differences in the definition of the neighborhood between the trustworthiness measure and cost function of NeRV. The neighborhood in the trustworthiness measure is defined as a step function instead of a smooth continuous function that covers all the data, like in NeRV. Moreover, the trustworthiness measure does not care about what happens to the points that are correctly inside the neighborhood. Thus NeRV uses resources in reducing errors that the trustworthiness measure does not care about. This also explains why selecting $\lambda = 0$ or $\lambda = 1$ is not always the best choice to maximize trustworthiness or continuity, respectively.

5 Demonstrations

To illustrate how the λ affects the NeRV results in practice, we used a difficult demonstration data set. The points are sampled uniformly from the surface of a three-dimensional sphere, and they are

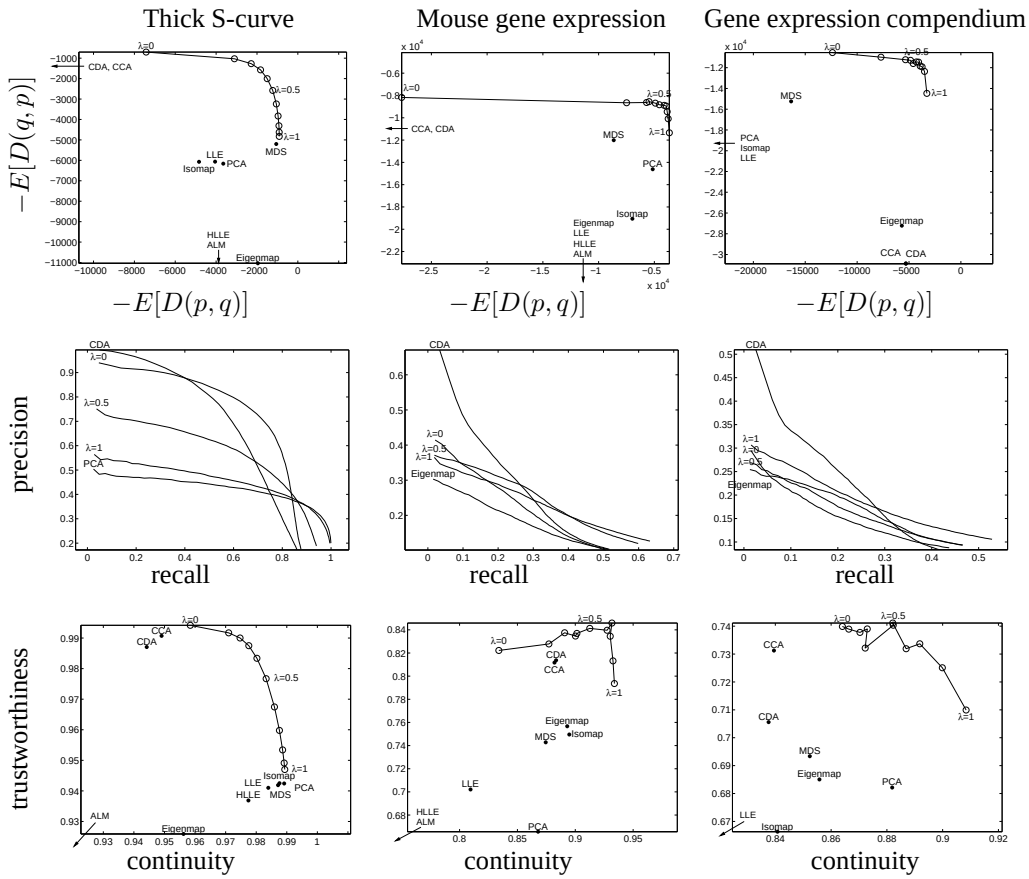

Figure 1: KL-KL curves (top), precision–recall curves (middle) and trustworthiness–continuity curves (bottom) for different values of λ on three data sets. Other nonlinear projection methods have been added for reference. The precision–recall curves have been calculated with 20 nearest neighbors in the input space as the set of relevant items and the number of retrieved items (neighbors) is varied from 1 to 100. Only the reference methods that achieved the highest precision and the highest recall are included for clarity. The KL–KL curve and the trustworthiness-continuity curve are calculated using 20 nearest neighbors. On each plot the best performance is in the top right corner. PCA: Principal Component Analysis, MDS: metric MultiDimensional Scaling, LLE: Locally Linear Embedding, Eigenmap: Laplacian Eigenmap, CCA: Curvilinear Component Analysis, CDA: CCA using geodesic distances, HLLE: Hessian Eigenmap, ALM: Alignment of Local Models.

to be projected to two dimensions. A perfect mapping is naturally impossible, and a compromise is needed. Figure 2 illustrates that NeRV with small value of λ cuts the sphere open to avoid false positives, resembling a geographical projection, whereas for large λ the sphere is squashed flat to minimize misses, resembling a linear projection. Note that the latter extreme corresponds to SNE, whereas the former resembles the results of CCA and CDA.

As a more practical demonstration we visualized the documents from the NIPS 2001–2003 conferences. The data vectors are histograms of log word counts in the documents, available from http://ai.stanford.edu/∼gal/. Such a display, optimized to maximize performance in retrieval of similar documents, could in practice be used by conference organizers to interactively define sessions and to choose referees. In both tasks a lot of prior knowledge naturally needs to be used, and the visualization might help in grasping the big picture, compared to the alternative of using pure information retrieval without such visualizations.

The four sub-images on the left side of Figure 3 show that the NIPS tracks are reasonably well clustered on the display, providing evidence that the overall order is probably sensible. We finally tested the system by mapping this paper to the display, shown on the right. The paper is in a very

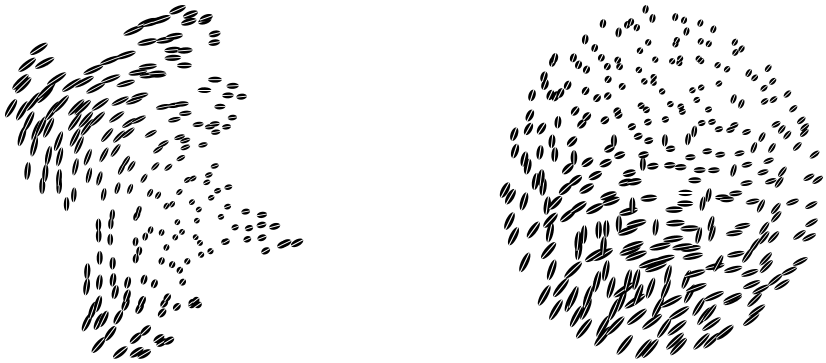

Figure 2: Two nonlinear projections of data that lies on the surface of a sphere to two dimensions. One of the input coordinates governs the rotation of the glyphs, the second their scale, and the third their degree of elongation. As a result, similarity of the glyphs indicates that the corresponding points are close to each other in the input space. On the *left*, $\lambda = 0$, the sphere has become split open and the glyphs change smoothly, but on the opposite ends of the projection there are similar glyphs that are projected far from each other. On the *right* $\lambda = 1$, the sphere has been squashed flat. There are areas where the different kinds of glyphs are close to each other, but there are no areas where similar glyphs are very far from each other. Only a small portion of the points used for computing the mapping are shown for clarity.

relevant area, containing many of the information visualization papers earlier presented in NIPS. It lies some distance away from the mass, however. It remains to be seen whether the distance is an indication of a significant new result or of something else.

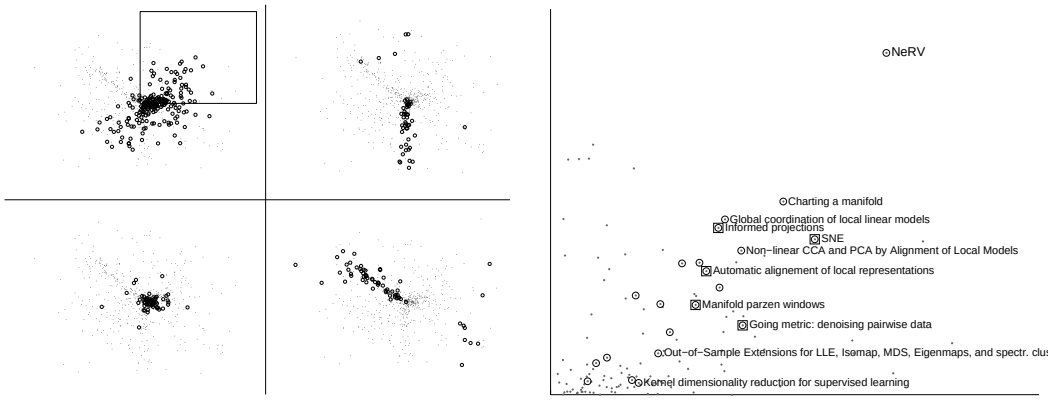

Figure 3: Documents from the NIPS 2001–2003 conferences projected onto a plane. On the left side, documents from one of the NIPS tracks are highlighted by circles in each sub-image, and the rest of the documents are marked by dots. Top left: Algorithms and architectures; Top right: Control and reinforcement learning, Bottom left: Learning theory; Bottom right: Neuroscience. Right half: Enlarged view of the area marked by the rectangle on the left. This area contains most of the documents related to dimensionality reduction, manifold learning or information visualization (circles); titles of a few documents are shown. The five papers that were presented in the 2002 spotlight session on dimensionality reduction and manifold learning are marked with squares. The current paper is in the top right corner; it has been marked with the label NeRV.

6 Discussion

We have introduced a new rigorous principle for optimizing nonlinear projections. The visualization task was conceptualized as neighbor retrieval, formulated as an information retrieval problem. The cost function measures the total cost of misses and false positives. We introduced an algorithm called NeRV (Neighbor Retrieval Visualizer) that extends the earlier Stochastic Neighbor Embedding method.

NeRV outperformed alternatives clearly for all three data sets we tried, and for two cost functions. By the third cost function NeRV was among the best but not a clear winner.

The weak point of NeRV is that it is computationally demanding. We have not analyzed the convergence properties but each gradient step is $\mathcal{O}(N^3)$ where N is the number of data points, whereas some of the alternatives are only $\mathcal{O}(N^2)$. Moreover, the cost function has local minima, in contrast to some alternatives which have a convex cost function.

In this paper we have assumed that the user chooses the compromise between precision and recall, governed by the parameter λ. It would be nice to be able to optimize it, but the optimization would increase the computational complexity. Choice of the value of λ is not a critical step, however, since NeRV outperforms the alternatives on a wide range of its values.

The visualizations could still be improved for interactive use with a simple additional visualization method. Given that the user selects a point on a display, the false positives in its neighborhood, and misses that are farther apart could be highlighted by special symbols.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS 14*, pages 585–591, Cambridge, MA, 2002. MIT Press.

[2] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer, New York, 1997.

[3] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154, 1997.

[4] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*, 100:5591–5596, 2003.

[5] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *NIPS 15*, pages 833–840. MIT Press, 2002.

[6] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441,498–520, 1933.

[7] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4:48, 2003.

[8] J. A. Lee, A. Lendasse, and M. Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49–76, 2004.

[9] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[10] E. Segal, N. Friedman, D. Koller, and A. Regev. A module map showing conditional activity of expression modules in cancer. *Nature Genetics*, 36:1090–1098, 2004.

[11] A. I. Su, M. P. Cooke, K. A. Ching, Y. Hakak, J. R. Walker, T. Wiltshire, A. P. Orth, R. G. Vega, L. M. Sapinoso, A. Moqrich, A.Patapoutian, G. M. Hampton, P. G. Schultz, and J. B. Hogenesch. Large-scale analysis of the human and mouse transcriptomes. *PNAS*, 99:4465–4470, 2002.

[12] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[13] J. J. Verbeek, S. T. Roweis, and N. Vlassis. Non-linear CCA and PCA by alignment of local models. In S. Thrun, L. Saul, and B. Schöllkopf, editors, *NIPS 16*, Cambridge, MA, 2004. MIT Press.