Nonlinear Dimensionality Reduction as Information Retrieval

Jarkko Venna and **Samuel Kaski**
Helsinki Institute for Information Technology
Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, FI-02015 TKK, Finland
jarkko.venna@tkk.fi, 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. A main application for both is in information visualization, to make visible the neighborhood or proximity relationships in the data, but neither approach has been designed to optimize this task. We give such visualization a new conceptualization 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, for which we give a new interpretation: it optimizes recall. The new method is shown empirically to outperform existing dimensionality reduction methods.

1 INTRODUCTION

Early nonlinear projection methods introduced a representation for the data points and optimized the representations to minimize representation error. Most can be interpreted as multidimensional scaling (MDS) methods (Borg & Groenen, 1997) 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 (Tenenbaum et al., 2000) infers the manifold through local neighborhood relationships, and visualizes it by MDS; Locally Linear Embedding (LLE; Roweis & Saul, 2000) approximates the manifold locally by linear surfaces; Laplacian Eigenmap (LE; Belkin & Niyogi, 2002) and Hessian Eigenmap (HLLE; Donoho & Grimes, 2003), are very similar but based on graph theory; Semidefinite Embedding (SDE; Weinberger et al., 2004) aims at maximizing the variance in the feature space while preserving the angles and distances between neighbors; Alignment of Local Models (ALM; Verbeek et al., 2004) 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 (CCA; Demartines & Hérault, 1997).

Nonlinear dimensionality reduction methods are commonly used for two purposes: (i) as preprocessing methods to reduce the number of input variables or to represent the inputs in terms of more natural variables describing the embedded data manifold, or (ii) for making the data set more understandable, by making the similarity relationships between data points explicit through visualizations. The visualizations are commonly needed in exploratory data analysis, and in interfaces to high-dimensional data. In this paper we will focus on the latter types of applications and call them *information visualization*, with the understanding that the goal is to visualize neighborhood or proximity relationships within a set of high-dimensional data samples. The introduced methods are expected to be useful for other kinds of dimensionality reduction tasks as well, however.

In information visualization applications, a problem with all the existing dimensionality reduction methods listed above is that they do not optimize the performance for the task of visualizing similarity relationships. The cost functions measure preservation of pairwise distances for instance, but that is 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 information 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; then 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 the users goals, 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 information visualization.

It turns out (derived below) that one of the manifold extraction methods, Stochastic Neighbor Embedding (SNE; Hinton & Roweis, 2002), 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 parameterized 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 DIMENSIONALITY REDUCTION AND INFORMATION RETRIEVAL

2.1 BACKGROUND: STOCHASTIC NEIGHBOR EMBEDDING

The SNE algorithm (Hinton & Roweis, 2002) 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 KL divergence D between the probability distributions in the input and output spaces, averaged over all points. The cost function is

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

where E_i is the average over data samples i.

2.2 SNE AS AN INFORMATION RETRIEVAL METHOD

SNE was originally motivated differently; we will next give it a new interpretation. It turns out that SNE can be seen as an information retrieval algorithm; it optimizes a smoothed form of recall. 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

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

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

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

where r is the neighborhood size in the output space.

Now each KL divergence in the cost function can be divided into four parts,

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

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 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}

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

and, moreover,

$$\begin{aligned} & D(p_i, q_i) \approx \\ & \left(N_{MISS} \frac{1-\delta}{k} - N_{FP} \frac{\delta}{N-k-1} \right) \log \frac{(1-\delta)}{\delta} \\ & \approx N_{MISS} \frac{1-\delta}{k} \log \frac{(1-\delta)}{\delta} = \frac{N_{MISS}}{k} C, \end{aligned} \tag{8}$$

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

$$\text{recall} = \frac{N_{TP}}{k} = 1 - \frac{N_{MISS}}{k}. \tag{9}$$

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 smoothed recall which 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 up new avenues for improvement. SNE maximizes (smoothed) recall, and it is well known that maximizing recall typically leads to low precision. In other words, SNE only optimizes one end of the spectrum.

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

$$D(q_i, p_i) \approx \frac{N_{FP}}{r} C, \tag{10}$$

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

$$\text{precision} = 1 - \frac{N_{FP}}{r}. \tag{11}$$

Hence, by reversing the direction of the KL 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 assign a relative cost λ to misses and $(1-\lambda)$ to false positives, then the total cost function to be optimized is

$$\begin{aligned} E_{\text{NeRV}} &= \lambda E_i[D(p_i, q_i)] + (1-\lambda) E_i[D(q_i, p_i)] \\ &= \lambda \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} + (1-\lambda) \sum_i \sum_{j \neq i} q_{ij} \log \frac{q_{ij}}{p_{ij}}. \end{aligned} \tag{12}$$

For step functions and small δ this reduces to a total information retrieval cost, a compromise between precision and recall, and for Gaussian functions as in SNE it can be interpreted as a smoothed cost. We call the new method that optimizes (12) *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 to focus more on either 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 algorithm. A heuristic but very effective way of avoiding local minima is to initialize the optimization by starting with a large width of the Gaussian neighborhood, 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 for each data point.

The computational cost of the gradient step in the NeRV algorithm is of complexity $\mathcal{O}(n^3)$, which can be prohibitive for large data sets. We will next sketch two alternative approximate cost functions with complexity $\mathcal{O}(n^2)$, to be compared empirically.

Zhu and Rohwer (1995) have developed an information geometric extension of the KL divergence that is valid for all positive measures instead of only normalized measures. The divergence is

$$D_{KLe}(p, q) = \sum q - p + p \log \frac{p}{q}. \tag{13}$$

Replacing the KL divergence in the NeRV cost function with the extension allows us to use the exponential density values without the soft-max normalization used in SNE and NeRV; this reduces the complexity of the gradient step. Furthermore, the change in the cost function makes it possible to replace conjugate gradients with stochastic gradient steps in the beginning of optimization, which avoids local optima. We call this approximate version *fast NeRV (fNeRV)*.

The new algorithms NeRV and fNeRV will additionally be compared with a similarly motivated but more heuristic earlier algorithm called *Local MDS* (Venna & Kaski, 2006). The Local MDS algorithm focuses on preserving distances that lie within a certain area of influence around each data point both in the input and output space. A parameter λ controls whether the focus is more on the preservation of distances that are local in the input space ($\lambda = 0$) or in the output space ($\lambda = 1$).

4 EMPIRICAL COMPARISON

We compared the performance of NeRV with alternative methods on four data sets; the first is a small artificial low-dimensional set, the next two very high-dimensional real-world sets, and the fourth is a.very high dimensional partly artificial set.

4.1 DATA SETS

Thick S-curve. The first data set is a simple toy set sampled from a folded low-dimensional manifold, a three-dimensional S-shaped manifold 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 (Su et al., 2002). 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 (Su et al., 2002), to select the genes for visualization. Of the mouse genes clearly expressed (average difference in Affymetrix chips, AD $>$ 200) 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 (`http://dags.stanford.edu/cancer`; Segal et al., 2004). 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.

Faces. The fourth data set is a selection of 698, synthetic images of a face (64x64 pixels). The pose and direction of lighting are changed in a system-

atic way to create a manifold in the image space. (`http://web.mit.edu/cocosci/isomap/datasets.html`; Tenenbaum et al., 2000). The raw pixel data was used.

4.2 METHODS

The performance of NeRV was compared with the following dimensionality reduction methods: Principal Component Analysis (PCA; Hotelling, 1933), metric Multidimensional Scaling (MDS; Borg & Groenen, 1997), Locally Linear Embedding (LLE; Roweis & Saul, 2000), Laplacian Eigenmap (LE; Belkin & Niyogi, 2002), Hessian Eigenmap (HLLE; Donoho & Grimes, 2003), Isomap (Tenenbaum et al., 2000), Alignment of Local Models (ALM; Verbeek et al., 2004), Curvilinear Component Analysis (CCA; Demartines & Hérault, 1997) and Curvilinear Distance Analysis (CDA; Lee et al., 2004), which is a variant of CCA that uses graph distances to approximate the geodesic distances in the data. LLE, LE, HLLE and Isomap were computed with code from their developers; MDS, ALM, CCA and CDA use our code.

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 several times with different random initializations and the best run was selected. For the NeRV and local MDS we set the (effective) number of neighbors k to 20 (without optimizing it further). A very small value will diminish the effect λ has on the tradeoff. This value was also used for fNeRV on all other data sets except for the Gene expression compendium data set for which the effective number of neighbors was set to 50. The smaller value caused the fNeRV algorithm to be stuck in local minima on this data set. In each case Euclidean distances were used in the input space.

4.3 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 of all methods on the plane spanned by the two measures (Fig. 1 left column). NeRV, fNeRV and local MDS form a curve parameterized by λ. NeRV was clearly the best-performing method on all four data sets, on this pair of measures. Local MDS and fNeRV have a relatively good smoothed precision but do not perform as well in terms of the smoothed recall, whereas Laplacian Eigenmap seems to be consistently good in this regard. Plain MDS is consistently a relatively good method as well.

Although the NeRV cost function is arguably a measure worth optimizing, we verified the results with two other pairs of performance measures. Since our motivation comes from information retrieval we 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 but is less sensitive to outliers on the other hand, and small local errors on the other. Trustworthiness (Kaski et al., 2003) 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 shown in the middle row of Figure 1. The CDA algorithm performed very well in terms of precision, being the best or second best on all four data sets. NeRV, fNeRV and local MDS perform well with a wide range of λ.

The trustworthiness and continuity measures (Fig.1, rightmost column) result mostly in similar conclusions as the KL plots in very leftmost column. One difference is that the highest trustworthiness for NeRV and fNeRV was often gained with λ in the middle of the scale. An explanation for this could be the differences in the definition of the neighborhood between the trustworthiness measure and the cost functions of NeRV and fNeRV. 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. Another difference is that both local MDS and fNeRV seem to have a better performance when measured with the continuity measure than with the KL divergence. The KL divergence is sensitive to situations where the output probability is close to zero when the input probability is not. This situation can easily happen with fNeRV and local MDS when λ is set close to zero.

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

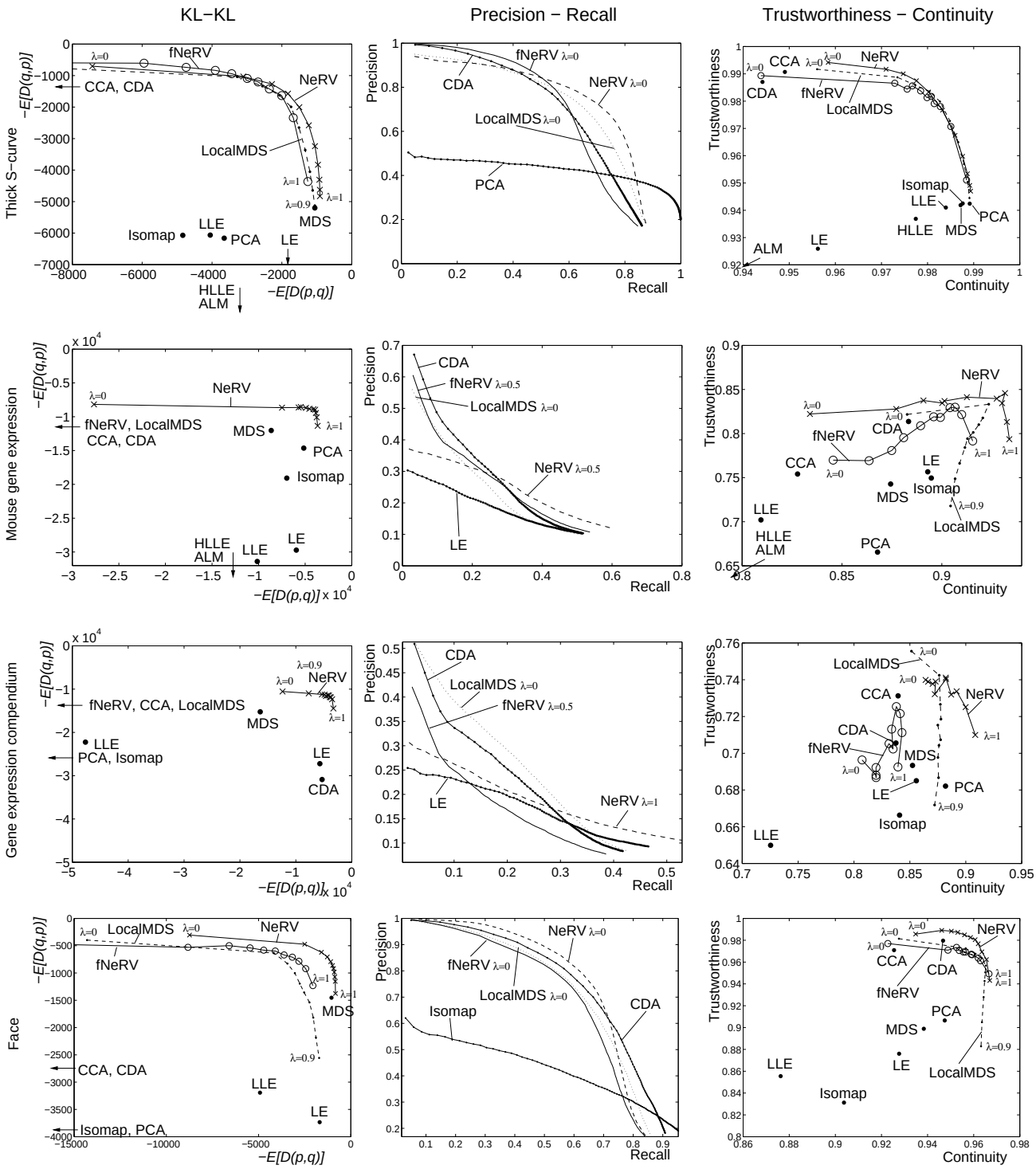

Figure 1: KL-KL curves (left), precision–recall curves (middle) and trustworthiness–continuity curves (right) for different values of λ on four data sets. Other nonlinear projection methods have been added for reference. The precision–recall curves have been calculated with 20 nearest neighbors in the 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 and the λ values that had the largest area under curve 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, LE: Laplacian Eigenmap, CCA: Curvilinear Component Analysis, CDA: CCA using geodesic distances, HLLE: Hessian Eigenmap, ALM: Alignment of Local Models.

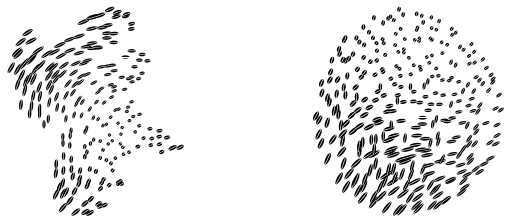

Figure 2: Two nonlinear projections of data that lies on the surface of a three-dimensional sphere. One of the input coordinates governs the rotation of the glyphs, the second their scale, and the third their degree of elongation. As a result, similarity of the glyphs indicates that the corresponding 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.

To further test how well the methods were able to recover the neighborhood structure inherent in the data we studied the face data set more closely. The true parameters for the pose and lighting used to generate the images in the data set are available. These parameters define the manifold embedded in the very high dimensional image space. We calculated the trustworthiness–continuity curves and the precision–recall curves using Euclidean distances in the pose and lighting space as the ground truth. In spite of the very high dimensionality of the input space and the reduction of the manifold dimension from three to two the results in Figure 3 show that NeRV, fNeRV and Local MDS were able to recover the structure well. The overall best trustworthiness was gained with NeRV ($\lambda = 0.1$) followed by CDA. The best continuity was gained by MDS followed by Laplacian Eigenmap.

5 DISCUSSION

We have introduced a new rigorous principle for optimizing nonlinear projections. The task of nonlinear projection for information visualization was conceptualized as neighbor retrieval and 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 four data sets we tried, and for two goodness criteria. By the third criterion NeRV was among the best but not a clear winner. Many of the popular manifold extraction methods perform surprisingly badly. The reason is that they have not been designed to reduce the dimensionality below the intrinsic dimensionality of the data manifold.

The weak point of NeRV is that it is computationally demanding. We constructed an approximate method by slightly changing the cost function; the resulting method fast NeRV (fNeRV) was empirically among the best performing methods although not as good as NeRV. Also our earlier similarly motivated but more heuristic method local MDS was comparable to or even better than fNeRV, and could hence be recommended as an alternative to it.

An implementation of the NeRV, fNeRV and local MDS algorithms, and of the trustworthiness and continuity measures is available at `http://www.cis.hut.fi/projects/mi/software/dredviz`

Acknowledgments

The authors belong to the Adaptive Informatics Research Centre, a national CoE of the Academy of Finland. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors views. All rights are reserved because of other commitments.

References

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS 14* (p. 585-591).

Borg, I., & Groenen, P. (1997). *Modern multidimensional scaling.* New York: Springer.

Demartines, P., & Hérault, J. (1997, January). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, *8*(1), 148–154.

Donoho, D. L., & Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*, *100*, 5591-5596.

Hinton, G., & Roweis, S. (2002). Stochastic neighbor embedding. In *NIPS 15* (pp. 833–840).

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

Kaski, S., Nikkilä, J., Oja, M., Venna, J., Törönen, P., & Castrén, E. (2003). Trustworthiness and met-

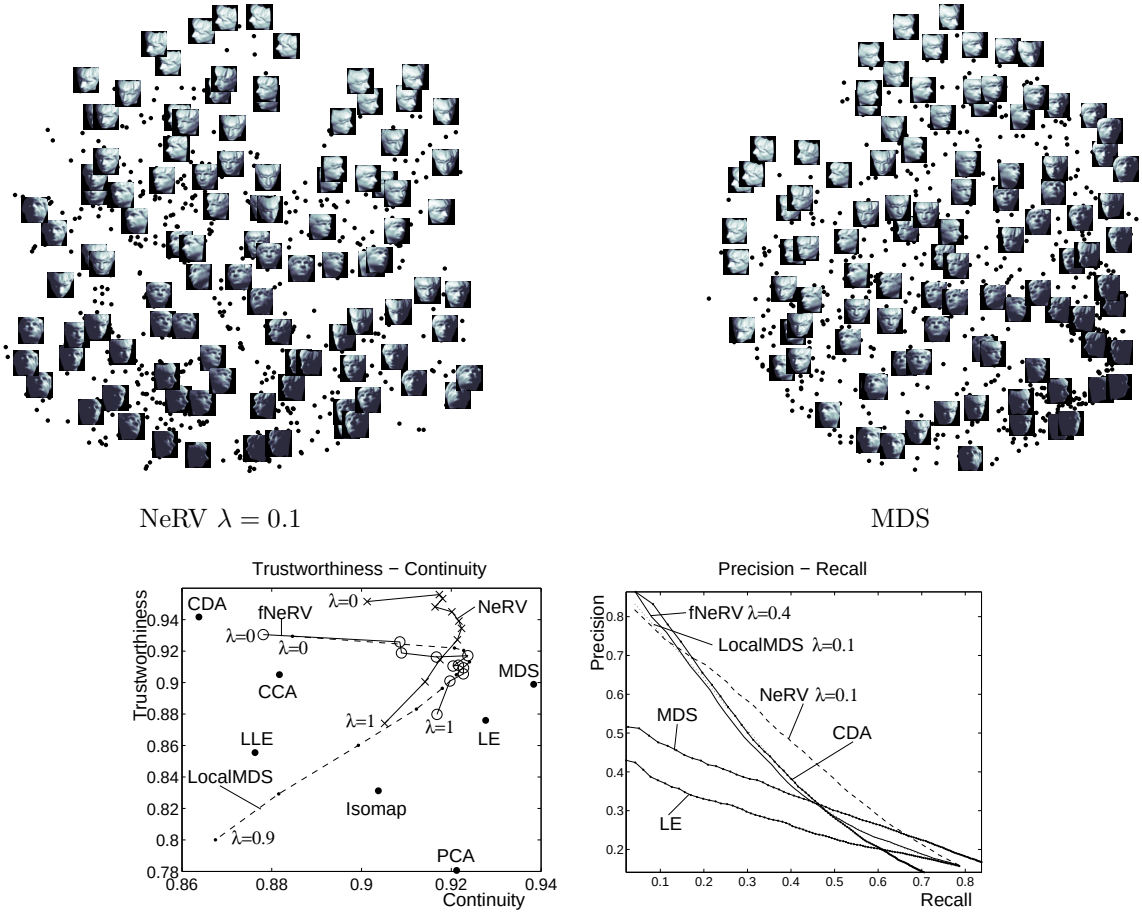

Figure 3: Top: Sample projections of the Face data set (NeRV vs. the best alternative). Bottom: How well was the original manifold reconstructed from the image data. Trustworthiness–continuity curves (left). The curves have been calculated using the known pose and lighting information as the input data. Neighborhood size is set to 20. Precision–recall curves (right). The curves have been calculated with 20 nearest neighbors in the known pose and lightning space as the set of relevant items, and the number of retrieved items (neighbors) is varied from 1 to 100. Only the best performing reference methods have been included for clarity, Key: see Figure 1

rics in visualizing similarity of gene expression. *BMC Bioinformatics*, *4*, 48.

Lee, J. A., Lendasse, A., & Verleysen, M. (2004). Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, *57*, 49–76.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Segal, E., Friedman, N., Koller, D., & Regev, A. (2004). A module map showing conditional activity of expression modules in cancer. *Nature Genetics*, *36*, 1090-1098.

Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., et al. (2002). Large-scale analysis of the human and mouse transcriptomes. *PNAS*, *99*, 4465-4470.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Venna, J., & Kaski, S. (2006). Local multidimensional scaling. *Neural Networks*, *19*, 889–899.

Verbeek, J. J., Roweis, S. T., & Vlassis, N. (2004). Non-linear CCA and PCA by alignment of local models. In *NIPS 16 (pp. 297–304).*

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the 21th international conference on machine learning (ICML'04)* (p. 106).

Zhu, H., & Rohwer, R. (1995). *Information geometric measurement of generalization* (Tech. Rep. No. NCRG/4350). Neural Computing Research Group, Aston University.