Visualization by Linear Projections as Information Retrieval

Jaakko Peltonen

Helsinki University of Technology, Department of Information and Computer Science,
P. O. Box 5400, FI-02015 TKK, Finland
jaakko.peltonen@tkk.fi

Abstract. We apply a recent formalization of *visualization as information retrieval* to linear projections. We introduce a method that optimizes a linear projection for an information retrieval task: retrieving neighbors of input samples based on their low-dimensional visualization coordinates only. The simple linear projection makes the method easy to interpret, while the visualization task is made well-defined by the novel information retrieval criterion. The method has a further advantage: it projects input features, but the input neighborhoods it preserves can be given separately from the input features, e.g. by external data of sample similarities. Thus the visualization can reveal the relationship between data features and complicated data similarities. We further extend the method to kernel-based projections.

但是它保留的输入邻域可以与输入特征分开给出

Key words: visualization, information retrieval, linear projection

1 Introduction

Linear projections are widely used to visualize high-dimensional data. They have the advantage of easy interpretation: each axis in the visualization is a simple combination of original data features, which in turn often have clear meanings. Linear projections are also fast to apply to new data. In contrast, nonlinear projections can be hard to interpret, if a functional form of the mapping is available at all. Some nonlinear methods also need much computation or approximation of the mapping to embed new points. Kernel-based projections are a middle ground between linear and nonlinear projections; their computation is linear in the kernel space, and their interpretability depends on the chosen kernel.

The crucial question in linear visualization is what criterion to use for finding the projection. Traditional answers include preservation of maximum variance as in principal component analysis (PCA); preservation of an independence structure as in independent component analysis; preservation of distances and pairwise constraints as in [1]; or maximization of class predictive power as in linear discriminant analysis, informative discriminant analysis [2], neighborhood components analysis [3], metric learning by collapsing classes [4], and others.

（PCA）中保留最大方差

在独立成分分析中保持独立结构

保留距离和成对约束

或最大化类预测能力

、邻域成分分析

通过折叠类进行度量学习

When the linear projection is intended for visualization, the previous criteria are insufficient, as they are only indirectly related to visualization. One must first formalize what is the task of visualization, and what are good performance measures for the task. This question has recently been answered in [5], where the task of visualization is formalized as an *information retrieval task*, and goodness measures are derived which are generalizations of *precision* and *recall*. Based on the goodness measures one can form an optimization criterion, and directly *optimize the goodness of a visualization in the information retrieval task*; however, so far this approach has only been used for nonlinear embedding where output coordinates are directly optimized without any parametric mapping [5, 6].

衡量好坏的标准就是经典的
基于上述的标准则可找到优化

We introduce a novel method for linear and kernel-based visualization called Linear Neighbor Retrieval Visualizer (LINNEA): we apply the formalization of visualization as an information retrieval task, and optimize precision and recall of such retrieval. A useful property is that the input features being projected and the distances used to compute the input neighborhoods *can be given separately*: for example, features can be word occurrence vectors of text documents and distances can be distances of the documents in a citation graph. In special cases, LINNEA is related to the methods *stochastic neighbor embedding* [7] and *metric learning by collapsing classes* [4], but it is more general; it can be used for unsupervised and supervised visualization, and allows the user to set the tradeoff between precision and recall of information retrieval. We show by preliminary experiments that LINNEA yields good visualizations of several data sets.

线性邻域检索可视化
器
将可视化的形式化应用为信息检索任务，
并优化此类检索的精度和召回率

2 Visualization as Information Retrieval

We briefly summarize the novel formalization of visualization introduced in [5].

The task is *visualization of neighborhood or proximity relationships* within a high-dimensional data set. For a set of input points $\mathbf{x}_i \in \mathbb{R}^{d_0}$, $i = 1, \ldots, N$, a visualization method yields output coordinates $\mathbf{y}_i \in \mathbb{R}^d$, which should reveal the neighborhood relationships. This is formalized as an *information retrieval task*: for any data point, the visualization should allow the user to retrieve its neighboring data points in the original high-dimensional data. Perfect retrieval from a low-dimensional visualization is usually not possible, and the retrieval will make two kinds of errors: not retrieving a neighbor decreases *recall* of the retrieval, and erroneously retrieving a non-neighbor decreases *precision*.

To apply the information retrieval concepts of *precision* and *recall* to visualization, in [5] they are generalized to continuous and probabilistic measures as follows. For each point i, a *neighborhood probability distribution* $p_{i,j}$ over all other points j is defined; in [5] an exponentially decaying probability based on input distances $d(\mathbf{x}_i, \mathbf{x}_j)$ is used. In this paper we allow the $d(\mathbf{x}_i, \mathbf{x}_j)$ to arise from any definition of distance between points i and j. The *retrieval of points from the visualization* is also probabilistic: for each point i a distribution $q_{i,j}$ is defined which tells the probability that a particular nearby point j is retrieved from the visualization. The $q_{i,j}$ are defined similarly to the $p_{i,j}$, but using Euclidean distances $||\mathbf{y}_i - \mathbf{y}_j||$ between visualization coordinates $\mathbf{y}_i$. This yields

对于每个点有一个 邻域概率分布
i 是这个点 j是其他点

基于输入距离的指数衰减概率d

$$p_{i,j} = \frac{e^{-d^2(\mathbf{x}_i,\mathbf{x}_j)/2\sigma_i^2}}{\sum_{k\neq i} e^{-d^2(\mathbf{x}_i,\mathbf{x}_k)/2\sigma_i^2}} \; , \qquad q_{i,j} = \frac{e^{-||\mathbf{y}_i-\mathbf{y}_j||^2/2\sigma_i^2}}{\sum_{k\neq i} e^{-||\mathbf{y}_i-\mathbf{y}_k||^2/2\sigma_i^2}} \qquad (1)$$

输入数据的邻域概率分布　　　　　映射（可视化后）的数据邻域概率分布

大概思路：
一般梯度下降算法是靠标签监督，这个方法将标签泛华成“距离”的一种，也就是一种嵌入化的标签。
在做可视化的时候可以根据关注的特征，灵活设置距离的定义，进而做到更符合需求情景的可视化。
具体来说，通过输入和结果的相对熵作为优化的指标，根据对precision和recall的重视程度设计消费函数，
找到输入到输出的线性变换参数矩阵W,构建一个输入到输出的映射。最后完成数据可视化的目标。
总体来说，和传统的机器学习算法类似，只是换成了嵌入可视化的角度，用一种更宏观且可操控的形式，进行
嵌入式降维

尺度参数，通过固定熵来设置p

用KL散度来衡量映射分布和输入分布的差异，置换p,q位置作precision和recall

这句话很关键

where σ_i are scale parameters which can be set by fixing the entropy of the $p_{i,j}$ as suggested in [5]. Since $p_{i,j}$ and $q_{i,j}$ are probability distributions, it is natural to use Kullback-Leibler divergences to measure how well the retrieved distributions correspond to the input neighborhoods. The divergence $D_{KL}(p_i, q_i) = \sum_{j \neq i} p_{i,j} \log(p_{i,j}/q_{i,j})$ turns out to be a *generalization of recall* and $D_{KL}(q_i, p_i)$ turns out to be a *generalization of precision*. The values of the divergences are averaged over points i which yields the final goodness measures.

基于两个指标，可以优化了但是通常不能同时优化它们，、做出取舍必须

3 The Method: Linear Neighborhood Retrieval Visualizer

The generalizations of precision and recall above can be directly used as optimization goals, but as both precision and recall cannot usually be maximized together, the user must set a tradeoff between them. Given the tradeoff a single cost function can be defined and visualizations can be directly optimized in terms of the cost function. In the earlier works [5, 6] this approach was used to compute a nonlinear embedding, that is, the output coordinates $\mathbf{y}_i$ of data points were optimized directly. In this paper we instead consider a parametric, linear projection $\mathbf{y}_i = \mathbf{W}\mathbf{x}_i$ where $\mathbf{W} \in \mathbb{R}^{d \times d_0}$ is the projection matrix. We wish to optimize $\mathbf{W}$ so that the projection is good for the information retrieval task of visualization. We call the method Linear Neighborhood Retrieval Visualizer (LINNEA). We use the same cost function as in [5], that is,

$$\begin{aligned} E &= \lambda \sum_i D_{KL}(p_i, q_i) + (1-\lambda) \sum_i D_{KL}(q_i, p_i) \\ &= \sum_i \sum_{j \neq i} \left[-\lambda p_{i,j} \log q_{i,j} + (1-\lambda) q_{i,j} \log \frac{q_{i,j}}{p_{i,j}} \right] + const. \end{aligned} \quad (2)$$

共轭梯度算法

where the tradeoff parameter λ is to be set by the user to reflect whether precision or recall is more important. We simply use a conjugate gradient algorithm to minimize E with respect to the matrix $\mathbf{W}$. The gradient $\frac{\partial E}{\partial \mathbf{W}}$ is

$$\sum_{i,j \neq i} \left[\lambda (p_{i,j} - q_{i,j}) + (1-\lambda) q_{i,j} \left(D_{KL}(q_i, p_i) - \log \frac{q_{i,j}}{p_{i,j}} \right) \right] \frac{(\mathbf{y}_i - \mathbf{y}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{\sigma_i^2} \quad (3)$$

which yields $O(N^2)$ computational complexity per gradient step.

通过初始化W作为主成分分析投影

Optimization details. In this paper we simply initialize the elements of $\mathbf{W}$ to uniform random numbers between 0 and 1; more complicated initialization, say by initializing $\mathbf{W}$ as a principal component analysis projection, is naturally possible. To avoid local optima, we use two simple methods. Firstly, in each run we first set the neighborhood scales to large values, and decrease them after each optimization step until the final scales are reached, after which we run 40 conjugate gradient steps with the final scales. Secondly, we run the algorithm from 10 random initializations and take the result with the best cost function value.

总体来说

3.1 Kernel Version

We now present a kernel version of LINNEA. Instead of simple linear projections we optimize projections from a kernel space: we set $\mathbf{y}_i = \mathbf{W}\Phi(\mathbf{x}_i)$ where $\Phi(\cdot)$ is some nonlinear transformation to a potentially infinite-dimensional space with inner products given by a kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$. As usual, the kernel turns out to be all we need and knowing Φ is not required.

The task is the same as before: to optimize the projection (visualization) so that it is good for information retrieval according to the cost function (2).

It is reasonable to assume that the rows $\mathbf{w}_l^T$ of $\mathbf{W}$ can be expressed as linear combinations of the $\Phi(\mathbf{x}_i)$, so that $\mathbf{w}_l = \sum_m a_m^l \Phi(\mathbf{x}_m)$ where a_m^l are the coefficients. Then the projection has the simple form

$$\mathbf{y}_i = \left[\sum_m a_m^1 \Phi(\mathbf{x}_m), \ldots, \sum_m a_m^d \Phi(\mathbf{x}_m)\right]^T \Phi(\mathbf{x}_i) = \mathbf{A}\mathbf{K}(\mathbf{x}_i) \tag{4}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{d \times N}$ contains the coefficients $A(l, m) = a_m^l$ and $\mathbf{K}(\mathbf{x}_i) = [k(\mathbf{x}_1, \mathbf{x}_i), \ldots, k(\mathbf{x}_N, \mathbf{x}_i)]^T$. As before, the coordinates $\mathbf{y}_i$ can be used to compute the neighborhoods $q_{i,j}$, the cost function, and so on.

为了优化这个基于内核的投影，
优化关于系数矩阵的成本函数
就足够了A。我们
可以再次使用标准的共轭梯度法：
关于A与等式（3）相同，除了X我
和Xj被替换为
K(x我） 和K(xj). 自从A有 否列，
计算复杂度变为在3个）每个梯度步骤。

To optimize this kernel-based projection, it is sufficient to optimize the cost function with respect to the coefficient matrix $\mathbf{A}$. We can again use a standard conjugate gradient method: the gradient with respect to $\mathbf{A}$ is the same as equation (3), except that $\mathbf{x}_i$ and $\mathbf{x}_j$ are replaced by $\mathbf{K}(\mathbf{x}_i)$ and $\mathbf{K}(\mathbf{x}_j)$. Since $\mathbf{A}$ has N columns, the computational complexity becomes $O(N^3)$ per gradient step.

3.2 Properties of LINNEA

A crucial property of LINNEA is that the input features $\mathbf{x}_i$ being projected and the distances $d(\mathbf{x}_i, \mathbf{x}_j)$ used to compute the input neighborhoods can be given separately. At simplest $d(\mathbf{x}_i, \mathbf{x}_j)$ can be the Euclidean distance $||\mathbf{x}_i - \mathbf{x}_j||$, but it can also be based on other data: for example, $\mathbf{x}_i$ can be word occurrence vectors of text documents and $d(\mathbf{x}_i, \mathbf{x}_j)$ can be distances of the documents in a citation graph (we test this example in Section 4). When distances are directly computed from input features, the projection is unsupervised. When distances are given separately the projection is supervised by the distances; then the projection is optimized to allow retrieval of neighbors which are based on the separately given distances, so it reveals the relationship between the features and the distances.

当直接从输入特征计算距离时，投影
是无监督的。
当距离单独给出时，投影由距离监督；
然后优化投影以允许检索基于
单独给定距离的邻居，因此它
揭示了特征与距离之间的关系。

综合了feature和距离

Note that a visualization based on the distances only, say multidimensional scaling computed from citation graph distances between documents, would not provide any relationship between the visualization and the features (document content); in contrast, the LINNEA visualization is directly a projection of the features, which is optimized for retrieval of neighbors based on the distances.

If we set $\lambda = 1$ in (2), that is, we maximize recall, this yields the cost function of *stochastic neighbor embedding* (SNE; [7]); thus LINNEA includes a linear version of SNE as a special case. More generally, the cost function of LINNEA implements a flexible user-defined tradeoff between precision and recall.

Another interesting special case follows if the input neighborhoods are derived from class labels of data points. Consider a straightforward neighborhood $p_{i,j}$: for any point i in class c_i, the neighbors are the other points from the same class, with equal probabilities. It is easy to show that if we set $\lambda = 1$ in the cost function (that is, we maximize recall), this is equivalent to maximizing

$$\sum_i \sum_{j \neq i} \delta_{c_i,c_j} \log \frac{e^{-||\mathbf{y}_i - \mathbf{y}_j||^2/2\sigma_i^2}}{\sum_{j' \neq i} e^{-||\mathbf{y}_i - \mathbf{y}_{j'}||^2/2\sigma_i^2}} \tag{5}$$

成本函数通过折叠类进行度量学习
以本文方法的角度解释已有的算法

where $\delta_{c_i,c_j} = 1$ if the classes (c_i, c_j) are the same and zero otherwise, and for simplicity classes are assumed equi-probable. This is the cost function of *metric learning by collapsing classes* (MCML; [4]) which was introduced as a supervised, linear version of SNE. LINNEA includes MCML as a special case. We thus give a new interpretation of MCML: it maximizes *recall of same-class points.*

Note that LINNEA yields meaningful solutions for the above kind of straightforward input neighborhoods because the mapping is a linear projection of input features. In contrast, methods that freely optimize output coordinates could yield trivial solutions mapping all input points of each class to a single output point. To avoid trivial solutions, such methods can e.g. apply topology-preserving supervised metrics as in [6]; such complicated metrics are not needed in LINNEA.

In summary, LINNEA can be used for both supervised and unsupervised visualization; it is related to well-known methods but is more general, allowing the user to set the tradeoff between the different costs of information retrieval.

4 Experiments

In this first paper we do not yet make thorough comparisons between LINNEA and earlier methods. We show the potential of LINNEA in four experiments; we use principal component analysis (PCA) as a baseline. We use the non-kernel version of LINNEA (Section 3), and set the user-defined tradeoff between precision and recall to $\lambda = 0$ (favoring precision only) in experiments 1-3 and $\lambda = 0.1$ in experiment 4. Other parameters were defaults from the code of [5].

Experiment 1: Extracting the relevant dimensions. We first test LINNEA on toy data where the visualization can perfectly recover the given input neighborhoods. Consider a spherical Gaussian cloud of 500 points in the Hue-Saturation-Value (HSV) color space, shown in Fig. 1 (left). One cannot represent all three dimensions in one two-dimensional visualization, and without additional knowledge one cannot tell which features are important to preserve, as the shape of the data is the same in all directions. However, if we are also given pairwise distances between points, they determine which features to preserve. Suppose those distances have secretly been computed based only on Hue and Value; then the correct visualization is to take those two dimensions, ignoring Saturation.

We optimized a two-dimensional projection with LINNEA; we gave the HSV components of each data point as input features, and computed input neighborhoods from the known pairwise distances. As shown in Fig. 1 (right), LINNEA

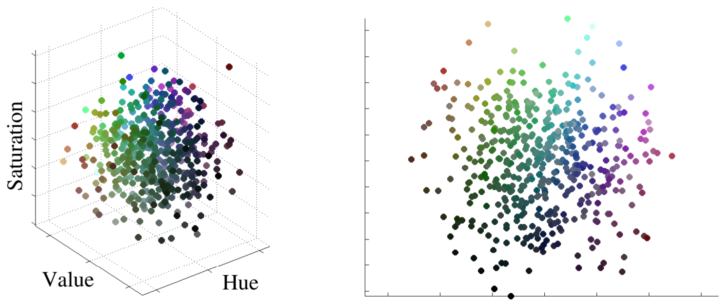

Fig. 1. Projection of points in the Hue-Saturation-Value color space. **Left:** the original three-dimensional data set is a Gaussian point cloud; coordinates correspond to Hue, Saturation (grayishness-colorfulness) and Value (lightness-darkness) of each dot. Pairwise input distances were computed from Hue and Value only. **Right:** LINNEA has correctly found the Hue and Value dimensions in the projection and ignored Saturation.

found the Hue-Value dimensions and ignored Saturation, as desired; the weight of Saturation in the projection directions is close to zero.

Experiment 2: S-curve. We visualize data set having a simple underlying manifold: 1000 points sampled along a two-dimensional manifold, embedded in the three-dimensional space as an S-shaped curve as shown in Fig. 2 (left). No external pairwise distances are given and input neighborhoods are computed from the three-dimensional input features. The task is to find a visualization where original neighbors on the manifold can be retrieved well. Unfolding the manifold would suffice; however, a linear projection cannot unfold the nonlinear S-curve perfectly. The PCA solution in Fig. 2 (middle) leaves out the original Z-axis, which is suboptimal for retrieving original neighbors as it leaves visible only one of the coordinates of the underlying two-dimensional manifold. The LINNEA result in Fig. 2 (right) emphasizes directions Z and Y; this shows the coordinates of the underlying manifold well and allows retrieval of neighbors of input points.

Experiment 3: Projection of face images. We visualize a data set of human faces ([8]; available at `http://web.mit.edu/cocosci/isomap/datasets.html`). The data set has 698 synthetic face images in different orientations and lighting directions; each image has 64×64 pixels. We first find linear projections of the face images using the pixel images as input features, without giving any additional knowledge. As shown in Fig. 3 (top left), PCA reveals part of the data structure, but the result is unsatisfactory for retrieving neighboring faces, since PCA has clumped together the back-lit faces. In contrast, as shown in Fig. 3 (top right), LINNEA spreads out both front-lit and back-lit faces. The projection directions can be interpreted as linear filters of the images. For PCA the filter on the horizontal axis roughly responds to a left-facing head; the filter

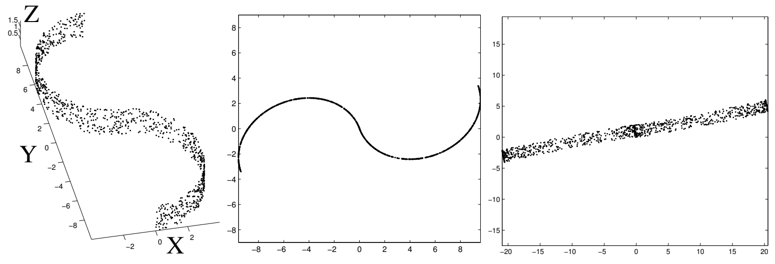

Fig. 2. Projections of an S-curve. **Left:** the original three-dimensional data. **Middle:** PCA neglects the original Z-direction. **Right:** LINNEA finds a projection where neighbors on the underlying manifold can be retrieved well from the visualization.

on the vertical axis roughly detects left-right lighting direction. The LINNEA filters are complicated; more analysis of the filters is needed in future work.

Projection to retrieve known pose/lighting neighbors. For the face data the pose and lighting parameters of the faces are available. We can then compute pairwise input distances based on these parameters, and use LINNEA to find a supervised visualization of the pixel images that best allows retrieval of the pose/lighting neighbors of each face. The LINNEA projection is shown in Fig. 3 (bottom). The face images are arranged quite well in terms of the pose and lighting; the top left–bottom right axis roughly separates left and right-facing front-lit faces, and the top right–bottom left axis roughly separates left and right-facing back-lit faces. The corresponding filters are somewhat complicated; the filters on the vertical axis and horizontal axis seem to roughly detect edges and lighting direction respectively. The underlying pose/lighting space is three-dimensional and cannot be represented exactly by a two-dimensional mapping, thus the filters are compromises between representing several aspects of pose/lighting. Note that running e.g. PCA on the known pose/lighting parameters would not yield filters of the pixel images, thus it would not tell how pixel data is related to pose/lighting; in contrast, LINNEA optimizes filters for retrieval of pose/lighting neighbors.

Experiment 4: Visualization of scientific documents. We visualize the CiteSeer data set which contains scientific articles and their citations. The data set is available at `http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html`. Each article is described by a binary 3703-dimensional vector telling which words appeared in the article; we used these vectors as the input features. To reduce computational load we took the subset of 1000 articles having the highest number of inbound plus outbound citations. We provide separate pairwise input distances, simply taking the graph distance in the citation graph: that is, two documents where one cites the other have distance 1, documents that cite the same other document have distance 2, and so on. As a simplification we assumed citation to be a symmetric relation, and as a regularization we upper bounded the graph distance to 10. We use LINNEA (with $\lambda = 0.1$) to optimize a two-dimensional visualization where neighbors according to this graph distance can

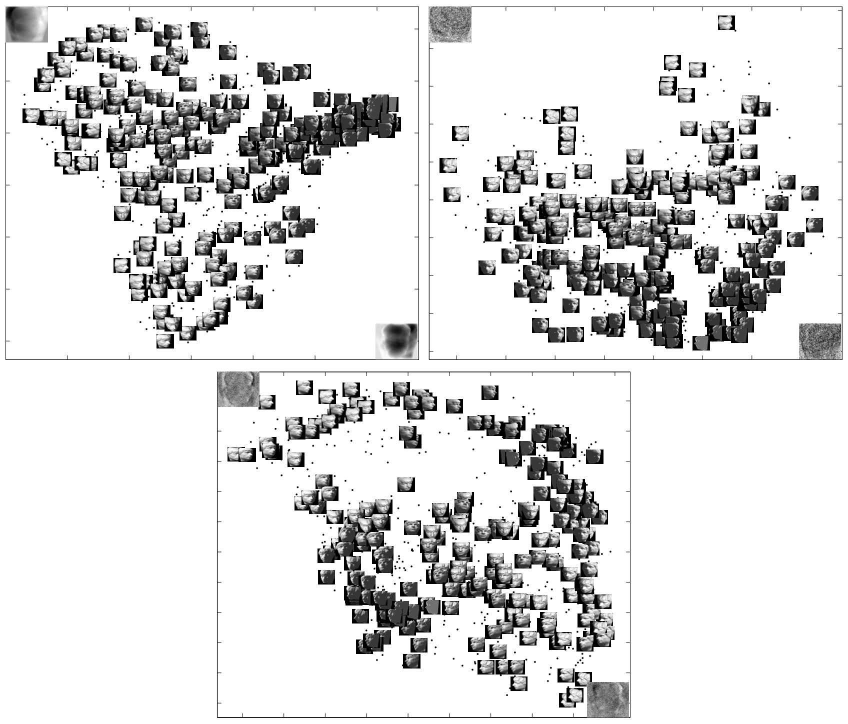

Fig. 3. Projections of face images. **Top:** unsupervised projections of the pixel images by PCA (left) and LINNEA (right). The linear projection directions can be interpreted as linear filters of the images, which are shown for each axis. **Bottom:** a supervised projection by LINNEA. Pairwise distances were derived from known pose/lighting parameters of the faces. LINNEA has optimized projections of the pixel images, for retrieving neighbors having similar pose/lighting parameters. See the text for more analysis.

be best retrieved. The result is shown in Fig. 4. In the baseline PCA projection (left subfigure) citations are spread all over the data with little visible structure, whereas the LINNEA projection (right subfigure) shows clear structure: clusters where documents cite each other, and citation connections between clusters. For this data each feature is a word; unfortunately the identities of the words are unavailable. In general one can interpret the projection directions given by LINNEA by listing for each direction the words having the largest weights.

5 Conclusions

We introduced a novel method for visualization by linear or kernel based projection. The projection is optimized for information retrieval of original neighbor points from the visualization, with a user-defined tradeoff between precision and recall. The method can either find projections for input features as such, or find

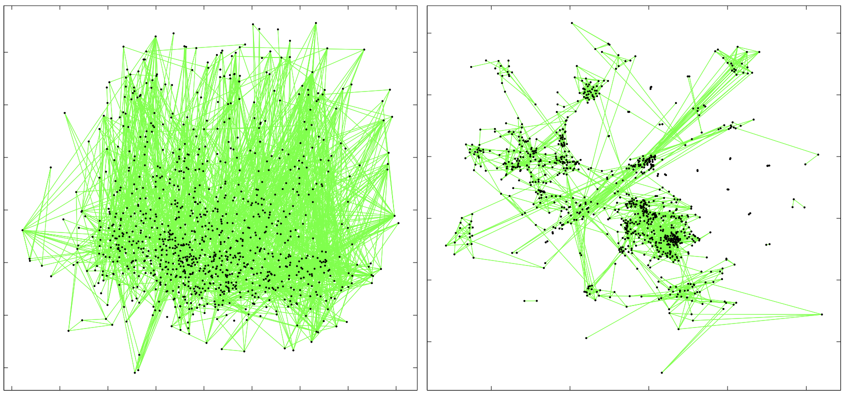

Fig. 4. Projections of scientific documents. Documents are shown as dots and citations between two documents are shown as lines. **Left:** PCA projection of document content vectors does not reveal citation neighborhoods well. **Right:** projection by LINNEA shows clusters of citing documents and connections between clusters.

projections that reveal the relationships between input features and separately given input distances. The method yields good visualization of several data sets.

Acknowledgments. The author belongs to the Adaptive Informatics Research Centre and to Helsinki Institute for Information Technology HIIT. He was supported by the Academy of Finland, decision 123983, and in part by the PASCAL2 Network of Excellence. He thanks Samuel Kaski for very useful discussion.

References

1. Cevikalp, H., Verbeek, J., Jurie, F., Kläser, A.: Semi-supervised dimensionality reduction using pairwise equivalence constraints. In: Proc. VISAPP 2008, pp. 489–496. (2008)
2. Peltonen, J., Kaski, S.: Discriminative components of data. IEEE Trans. Neural Networks **16**(1), 68–83 (2005)
3. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: Proc. NIPS 2004, pp. 513–520. MIT Press, Cambridge, MA (2005)
4. Globerson, A., Roweis, S.: Metric learning by collapsing classes. In: Proc. NIPS 2005, pp. 451–458. MIT Press, Cambridge, MA (2006)
5. Venna, J., Kaski, S.: Nonlinear dimensionality reduction as information retrieval. In: Proc. AISTATS*07. (2007)
6. Peltonen, J., Aidos, H., Kaski, S.: Supervised nonlinear dimensionality reduction by neighbor retrieval. In: Proc. ICASSP 2009. In press.
7. Hinton, G., Roweis, S.T.: Stochastic neighbor embedding. In: Proc. NIPS 2002, pp. 833–840. MIT Press, Cambridge, MA (2002)
8. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (December 2000)