Embedded Map Projection for Dimensionality Reduction-Based Similarity Search

Simone Marinai, Emanuele Marino, and Giovanni Soda

Dipartimento di Sistemi e Informatica - Università di Firenze
Via S.Marta, 3 - 50139 Firenze - Italy

Abstract. We describe a dimensionality reduction method based on data point projection in an output space obtained by embedding the Growing Hierarchical Self Organizing Maps (GHSOM) computed from a training data-set. The dimensionality reduction is used in a similarity search framework whose aim is to efficiently retrieve similar objects on the basis of the Euclidean distance among high dimensional feature vectors projected in the reduced space. This research is motivated by applications aimed at performing Document Image Retrieval in Digital Libraries. In this paper we compare the proposed method with other dimensionality reduction techniques evaluating the retrieval performance on three data-sets.

1 Introduction

In Pattern Recognition, objects to be recognized are frequently represented by collections of features that, when organized in vectors, allow us to represent objects as points in a vector space. The use of a suitable distance (or similarity measure) among objects can give rise to significant differences in the recognition performance achievable, but in most cases the Euclidean distance is used.

In this work we deal with Euclidean vector spaces. Our main application domain is in the field of Document Image Retrieval (DIR) where the aim is to identify relevant documents relying on image features only (e.g. considering layout-based retrieval or word indexing [1]). In this paper we will consider three datasets. In each collection the objects are represented as n-dimensional points and we aim to retrieve objects on a query by example paradigm: given a query object (n-dimensional point) we identify the most similar objects by looking for nearest points in the feature space. In principle the approach is simply based on an exhaustive comparison of the query point with all the indexed points, followed by a sorting of the computed distances. This strategy has severe limits for the computational cost involved and several approaches have been proposed to alleviate this problem.

From one side various multidimensional indexing methods have been proposed as extensions of spatial indexes such as quad-trees and R-trees [2]. Multidimensional methods (e.g. X-tree) are aimed at indexing high-dimensional data more efficiently than the sequential scan. When dealing with "very high dimensional" data (hundreds or thousands of dimensions) many multidimensional indexes tend

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 582–591, 2008.

to degenerate and perform poorly than the sequential scan. Some studies have been performed to clarify these poor performances that are generally attributed to the so called *curse of dimensionality* [3]. The curse of dimensionality has several facets. One of the most known is the property that *independent and identically distributed* points are mostly concentrated on the sides of a unit cube as long as the number of dimensions grows [4]. These problems become evident for dimensions as low as 10-15. However, distributions of points representing real objects are unlikely to be uniformly distributed and therefore similarity methods still work for higher dimensions.

The latter consideration lead to a group of approaches that adopt a dimensionality reduction of the data as a preliminary processing step, before using a multidimensional index on the reduced space (e.g. [5]). Working on a reduced space the quality of the query results can be reduced, giving rise to wrong results both in terms of false positives and negatives [2]. If the general aim is to identify *some* relevant objects at the risk of loosing other positive hits (similarly to a Web search engine) then this approach can be considered.

In this paper we describe a dimensionality reduction technique that we designed to deal with an image based word indexing in a DIR application. The proposed method is based on the use of Growing Hierarchical Self Organizing Maps (GHSOM) that cluster input vectors into a hierarchy of multiple layers consisting of several independent SOMs. The hierarchy is deepest in correspondence with more complex clusters, that are represented with more details by lower level maps. The GHSOM has been mainly used as a visualization method and in this case the various maps are usually explored independently one to each other. The peculiarity of our approach is that we embed the lower level maps in the root one so as to obtain an unique low dimensional space where input patterns are projected by interpolation with respect to the cluster centers.

The paper is organized as follows. In Section 2 we summarize the previous work on dimensionality reduction and we describe the basic characteristics of Self Organizing Maps that are useful to understand the proposed method described in Section 3. The comparison of our method with other dimensionality reduction methods is analyzed in Section 4, whereas our final remarks are drawn in Section 5.

2 Related Work

The literature on dimensionality reduction is large, summarizing a longstanding research. In this section we concentrate only on some characteristics of the methods that we have considered for comparison with our approach.

Principal Components Analysis (PCA) is one of the most popular linear techniques. PCA performs dimensionality reduction by embedding the data into a lower dimensional space finding a linear basis in which the variance in the data is maximal. We omit here the details of PCA, however it is important to point out that PCA is based on the computation of the covariance matrix of the input data and subsequent evaluation of the principal eigenvectors of this matrix, that

form the basis of the reduced space. After computing the PCA transformation, the mapping of points in the reduced space can be simply computed by means of a matrix multiplication and this is one of the main advantages of PCA-based dimensionality reduction.

In real data, often a linear mapping does not suffice to perform the dimensionality reduction and non-linear mappings should be considered. There are two main classes of algorithms: global and local techniques that attempt to preserve global (local) properties of the input data [6].

In the last few years deep architectures gained attention in the machine learning community. Autoencoders are a widely known architecture in this framework that have been used for global nonlinear dimensionality reduction since the 1990's [7]. Autoencoders are Multilayer perceptrons (MLP) having the same number of input and output units and a lower number of nodes in a hidden layer. The training is performed by means of the standard back-propagation algorithm where the network is forced to reproduce in the output layer the input patterns. After the training the hidden units are expected to describe the training data with a smaller representation, performing a non-linear dimensionality reduction. Similarly to other MLP-based architectures, autoencoders are prone to get stuck in local minima. To overcome this problem a new training strategy has been recently proposed [8]. The idea is to use Restricted Boltzmann Machines (RBM) for a preliminary unsupervised training of the recognition layers of the network. After the RBM training the reconstruction layers are formed by the inverse of the trained recognition layers. At the end of the process the overall autoencoder is fine tuned with the standard back-propagation algorithm. A trained autoencoder can be subsequently used to perform dimensionality reduction, by considering the output of the deeper recognition layer when presenting an input pattern in the original space.

Local Tangent Space Analysis (LTSA) is a local technique that is based on the representation of the local geometry of the manifold using tangent spaces learned by fitting an affine subspace in a neighborhood of each data point [9]. This local space is estimated by computing the PCA on the k nearest points of each input point. The tangent spaces are aligned so as to obtain the global coordinates of the data points with respect to the underlying manifold. This alignment is made by means of a partial eigendecomposition of the neighborhood connection matrix. One limitation of LTSA is the lack of an out-of-sample extension used to embed additional data points into an LTSA representation. It is therefore not possible to index additional objects or to perform queries with objects not indexed. This limit does not affect PCA, autoencoders and the method proposed in this paper.

2.1 Self Organizing Maps and GHSOM

The Self-Organizing Map (SOM) is a kind of artificial neural network that is based on unsupervised learning [10]. In the SOM the neurons are typically arranged in a two dimensional grid. Each neuron of the SOM is associated with a weight vector, or centroid, $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T \in \Re^n$. During learning, each

input vector $x \in \Re^n$ is compared with all the weight vectors of the SOM and the Best Match Unit (BMU) is determined: the BMU is the weight vector closest to the input vector according to a given distance, for instance the Euclidean distance. The BMU c is therefore defined by

$$c = arg \min_i \parallel x - w_i \parallel \tag{1}$$

where x is an input vector and w_i is the weight vector associated with neuron i. The input vector x is then mapped to the cluster represented by neuron c, and its weight vector and those of its neighbors are updated. In the update process, two types of rules may be used: the *incremental* learning rule and the *batch* learning rule.

The need to predefine the SOM structure results in a significant limitation on the final mapping achievable . To avoid the problem of the static SOM structure various dynamic algorithms have been proposed. One example is the Growing Hierarchical Self-Organizing Map (GHSOM) that dynamically models the training data [11]. The GHSOM allows the network structure to grow in two dimensions, in width and in depth, so that it combines the advantages of the dynamic growth and of the hierarchical structure. The growing process of the GHSOM is regulated by two parameters. The parameter τ_1 regulates the growing process in width while τ_2 regulates the growing process in depth. Each layer of the trained GHSOM consists of several independent SOMs.

The GHSOM training starts from the first layer with an initial 2x2 grid. After a pre-defined number of training iterations, the unit E with the highest *quantization error* is identified, and a row or a column of neurons is inserted between the unit E and its most dissimilar neighboring unit. After this insertion, the training restarts with the standard SOM algorithm. When the training of a map ends, each unit satisfying a predetermined condition is subjected to hierarchical expansion, therefore producing the hierarchical structure of the GHSOM.

3 Embedded Map Projection

As already discussed, real patterns are unlikely to belong to a uniform distribution in the original vector space. On the opposite, the patterns can frequently be imagined as laying to low dimensional manifolds or, in other cases, as belonging to clusters, that can be more or less elongated in the input pattern space. The latter feature has been considered to speed-up the indexing and retrieval of objects in high dimensional spaces. For instance, the Cluster Tree [12] is an index structure that has been proposed to perform approximate search in high dimensional spaces on the basis of pattern clustering. In [13] we combined the SOM clustering with the PCA to efficiently index words represented by points in high dimensional spaces. Words in each cluster, that are expected to be more similar one to the others, are projected into a lower dimensional space by means of a local PCA, and this reduced representation is used to speed up the similarity search.

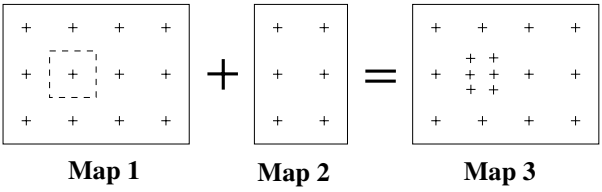

Fig. 1. Embedding of a second level SOM (Map2) into the parent one (Map1)

In [13] we did not consider the topological order of clusters because complex patterns, such as words, can not be easily modeled by a single map. One solution is to use a larger SOM, but training such map is not easy and the retrieval time risks to be very high, since the number of clusters quickly becomes very large. An alternative is to use a hierarchical map that is composed of several maps at different levels of detail such as the GHSOM. To have an idea of the type of maps that are built in the hierarchy, in the right part of Figure 3 we report the first level map and two sub-maps computed for the MNIST dataset.

The basic idea of the proposed approach is to embed lower maps in the output space that is implicitly defined by the first level map. Input points are then projected in this space. The dimensionality reduction is therefore a two step process: first, an embedding map is constructed starting from a trained GHSOM; second, input points are projected in this embedded map. Before describing the two steps, it is important to clarify that the GHSOM training, and subsequent embedded map building, is performed on a reduced number of points randomly selected from the collection to be indexed. The whole dataset is used in the projection step.

3.1 Embedded Map Building

To explain the concept we show in Figure 1 the embedding of a second order map (Map2) in the corresponding parent map (Map1). The SOM neurons (corresponding to cluster centers) can be considered as belonging to a two-dimensional grid, where each neuron is identified by two indexes. If we use the grid position as a low dimensional coordinate system, then the clusters correspond to uniformly distributed points represented by crosses in Map1. In the Figure, Map1 is a 3x4 map, whose cluster (1,1) (surrounded by a dashed box) is furtherly described with a 3x2 map (Map2). Embedding Map2 into Map1 we obtain Map3 that describes with more resolution the region in the output space corresponding to the neuron (1,1) in Map1. The actual embedding is obtained by a recursive linear scaling of lower level grids in the main one. A real embedded map computed with the MNIST dataset is shown in Figure 3.

3.2 Point Projection

As discussed in Section 2 the BMU for a pattern x represents the cluster having the highest similarity with x. By assigning x to this cluster we make a discrete

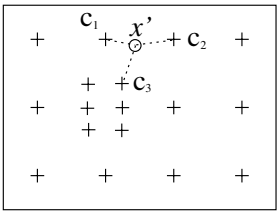

Fig. 2. Example of the projection of a point in the embedded map shown in Figure 1. Details are described in the text.

dimensionality reduction, where the output space is populated only in a limited number of fixed positions. If we aim at a less rigid projection, we can take advantage of the topological ordering of the map. Due to the curse of dimensionality, the quantization errors of nearest centroids have in general similar values, however we can consider these quantization errors as weights to be used to project the input point in a position in the embedded map that is between the closest centroids.

Previous approaches to perform this projection worked with single SOMs [14], or with each individual map in the GHSOM [15]. The projection described here deals with the embedded map centroids and projects each input point x as follows (see also Figure 2). Let c_1 be the BMU for point x and c_2 and c_3 be the next closest centroids with distances $d_i = \| x - c_i \|$ $(i = 1, 2, 3)$. The three distances are ordered so that $d_1 \leq d_2 \leq d_3$ and x should be placed somewhere between the three points, but closer to c_1. The projection is therefore defined by the following rule:

$$x' = \frac{d_1^{-1} c_1 + d_2^{-1} c_2 + d_3^{-1} c_3}{d_1^{-1} + d_2^{-1} + d_3^{-1}}. \tag{2}$$

It is important to remark that this projection has been already proposed [15], however in that case it was used for visualization of points in *separate* maps, whereas in our case we use it to project the data in a unique low dimensional space.

4 Experiments

In this section we describe the experiments performed on three datasets comparing four dimensionality reduction methods for similarity retrieval. The software used is implemented in Matlab. In particular the PCA, LTSA, and autoencoder software has been described in [16]. The proposed embedded map projection method has been implemented starting from the GHSOM Toolbox for Matlab [11].

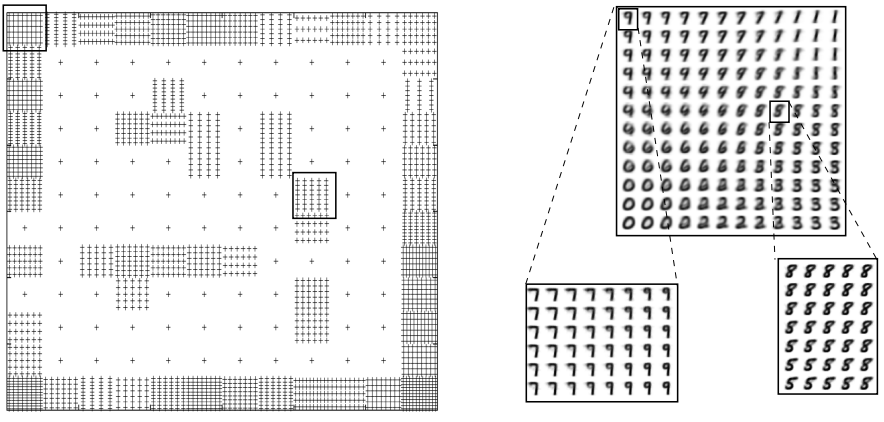

Fig. 3. The embedding obtained for the MNIST dataset. A part of the GHSOM is shown on the right. In particular we show the root map and two lower level maps (corresponding to the marked regions in the embedded map).

4.1 Datasets

The experiments have been performed on three datasets that represent various domains. The MNIST dataset is a widely used collection of handwritten digits containing 60,000 images with size of 28 x 28 pixels. The COIL20 dataset contains images of 20 objects, depicted from 72 different viewpoints each. Therefore, there are a total of 1,440 images that have a size of 32 x 32 pixels. The WORDS dataset is a collection of digitized printed words normalized to fit a 12 x 57 grid. In total we have 132,956 word images extrapolated from 1,302 pages that are part of an encyclopedia of the XIX^{th} Century.

At first, we projected the input data of each dataset into a two dimensional space. In the PCA we only needed to define the output dimension (2 in this case). For the training we used 10,000 patterns for MNIST, all the patterns for COIL20, and 13,296 patterns for WORDS. For MNIST and WORDS we subsequently projected all the patterns in the datasets. For the autoencoders we used for all the datasets a 1000-500-250-2 units in the hidden layers and 50 epochs for RBM training, whereas the fine-tuning was made with 100, 200, 300 back-propagation epochs retaining the best network. For the GHSOM training we evaluated several combinations of parameters, but for all the datasets the best results have been achieved with $\tau_1 = 0.6$ and $\tau_2 = 0.005$. The resulting maps have the following features[1]: MNIST(2,60,2566), COIL20(4,70,395), WORDS(2,60, 2077). For LTSA we set $k = 12$.

Figure 5 shows the projections for MNIST of the whole dataset and of the first seven classes separately. From the global map we can notice that the proposed EMP method distributes the patterns in the output space more uniformly and,

[1] The notation used is DATASET(# levels, # maps, # clusters).

Table 1. Precision at 0 percent Recall with the four compared methods

Dataset	n	N	EMP	PCA	Autoencoder	LTSA
MNIST	784	10,000	87.14	55.17	66.14	77.25
		60,000	84.56	53.55	64.66	–
COIL20	1024	1,440	85.59	71.68	82.35	19.63
WORDS	684	132,956	79.80	9.91	30.29	–

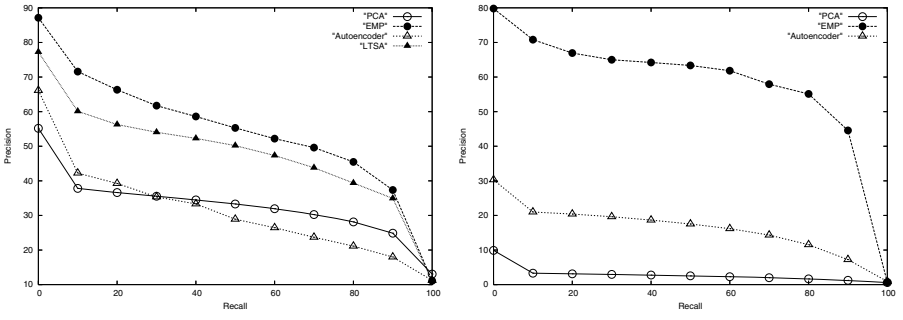

Fig. 4. Precision-Recall plot for the four compared methods working on the MNIST and WORDS data-sets

even if there are more outliers, in general the patterns of different classes are more separated with respect to the other methods.

4.2 Numerical Evaluation

The effectiveness of the methods has been measured in term of accuracy achieved by a query by example retrieval performed on the reduced space. We made several queries and we computed a Precision-Recall plot averaging each query. For the COIL20 dataset we used in turn each point as query evaluating the retrieval performance. For MNIST we used 10,000 queries randomly selected from the whole 60,000 patterns. In the case of the WORDS dataset there are only 576 labeled words that are used as queries.

Figure 4 shows the Precision-Recall plots for the MNIST and WORDS data-sets. In the MNIST dataset we projected only the 10,000 training patterns, and therefore we have also a plot for LTSA. In the WORDS dataset the LTSA plot is missing, since we indexed all the 132,956 points. To reduce the paper length we summarize in table 1 all the performed experiments reporting the Precision at Recall 0 for various experiments (this value is obtained by an interpolation of the Precision Recall plots). In the table, n is the dimension of the input space, N is the number of indexed objects.

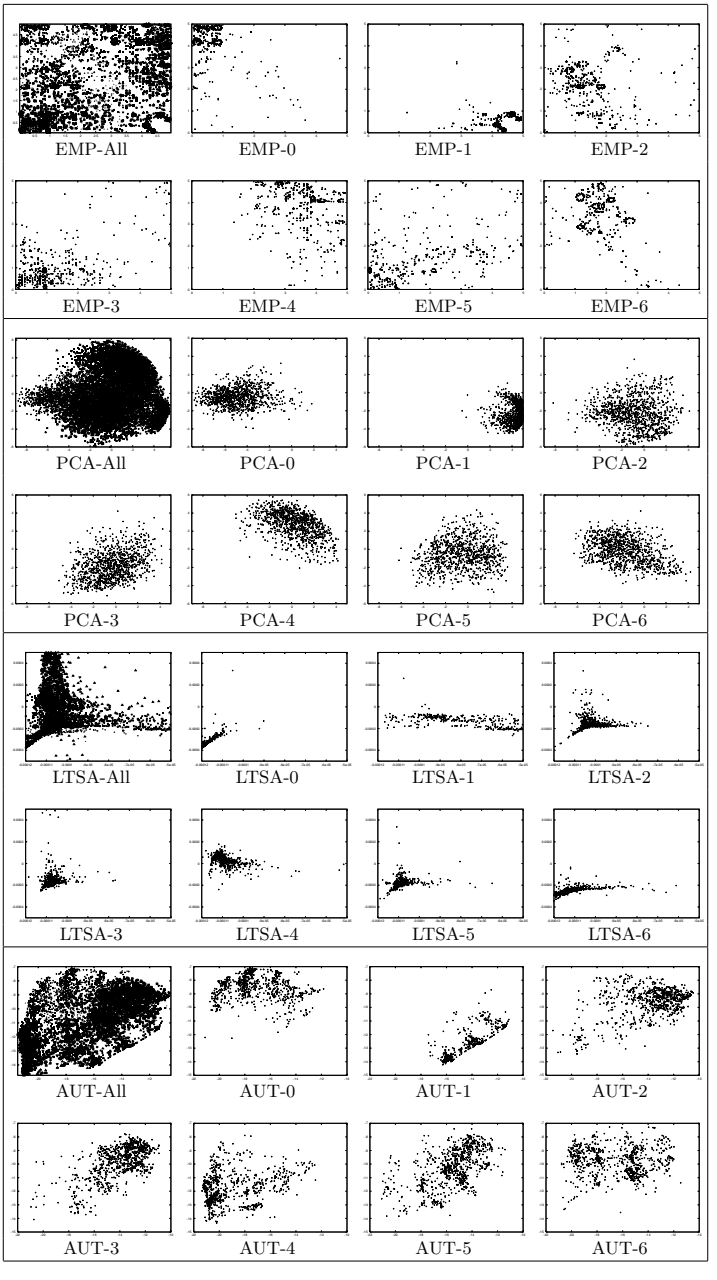

Fig. 5. Two dimensional projection of the MNIST dataset obtained with the four methods. For each method we report the distribution of points for the whole dataset, and for the first seven classes (0-6).

5 Conclusions

In this paper we propose a dimensionality reduction method that is based on the embedding of lower level maps of a GHSOM clustering of the input data. The method has been compared with other dimensionality reduction methods on a query by example retrieval application on three datasets. These preliminary results are encouraging, since the EMP method outperforms the compared ones.

References

1. Marinai, S., Marino, E., Soda, G.: Font adaptive word indexing of modern printed documents. IEEE Transactions on PAMI 28(8), 1187–1199 (2006)
2. Samet, H.: Foundations of multidimensional and metric data structures. Morgan Kaufmann, Amsterdam (2006)
3. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction. Springer series in statistics, New York, NY, USA (2001)
4. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Proc. 7th Int. Conf. on Database Theory, vol. 3, pp. 1763–1768 (1999)
5. Kanth, K.V.R., Agrawal, D., Singh, A.: Dimensionality reduction for similarity searching in dynamic databases. SIGMOD Rec. 27(2), 166–176 (1998)
6. van der Maaten, L., Postma, E., van den Herik, H.: Dimension reduction: A comparative review (preprint, 2007)
7. De Mers, D., Cottrell, G.: Nonlinear dimensionality reduction. In: NIPS-5 (1993)
8. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
9. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. SIAM Journal of Scientific Computing 26(1), 313–338 (2004)
10. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection. IEEE Transactions on Neural Networks 11(3), 574–585 (2000)
11. Chan, A., Pampalk, E.: Growing hierarchical self organising map (ghsom) toolbox: visualisations and enhancements. In: Neural Information Processing, ICONIP 2002. Proceedings of the 9th International Conference, vol. 5, pp. 2537–2541 (2002)
12. Li, C., Chang, E., Garcia-Molina, H., Wiederhold, G.: Clustering for approximate similarity search in high-dimensional spaces. IEEE Transactions on Knowledge and Data Engineering 14(4), 792–808 (2002)
13. Marinai, S., Faini, S., Marino, E., Soda, G.: Efficient word retrieval by means of SOM clustering and PCA. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 336–347. Springer, Heidelberg (2006)
14. Wu, Z., Yen, G.: A som projection technique with the growing structure for visualizing high-dimensional data. In: Proceedings of the International Joint Conference on Neural Networks, vol. 3, pp. 1763–1768 (2003)
15. Yen, G., Wu, Z.: Ranked centroid projection: a data visualization approach with self-organizing maps. IEEE Transactions on Neural Networks 19(2), 245–258 (2008)
16. van der Maaten, L.: An introduction to dimensionality reduction using matlab. Technical Report Technical Report MICC 07-07 (2007)