Robust Automatic Data Decomposition Using a Modified Sparse NMF

Oksana Samko, Paul L. Rosin, and A. Dave Marshall

School of Computer Science, Cardiff University, UK
{O.Samko, Paul.Rosin, Dave.Marshall}@cs.cf.ac.uk

Abstract. In this paper, we address the problem of automating the partial representation from real world data with an unknown a priori structure. Such representation could be very useful for the further construction of an automatic hierarchical data model. We propose a three stage process using data normalisation and the data intrinsic dimensionality estimation as the first step. The second stage uses a modified sparse Non-negative matrix factorization (sparse NMF) algorithm to perform the initial segmentation. At the final stage region growing algorithm is applied to construct a mask of the original data. Our algorithm has a very broad range of a potential applications, we illustrate this versatility by applying the algorithm to several dissimilar data sets.

1 Introduction

The objective of this paper is to present a novel automatic method for learning a meaningful sub-part representation from real world data with an unknown a priori structure.

Modelling the data sub-parts individually has great advantages. The underlying representation has proven to be accurate in representing the specificity of the data and capturing small but important variation in the data that are otherwise lost in standard approaches. For example in tracking human motion the hierarchy is naturally the key joints of the human skeleton suitably decomposed into the whole body, torso, arms legs upper arms etc. Adopting such a hierarchy and projecting down through the subspaces led to greater tracking accuracy in the model. In the talking head application, the hierarchy uses both visual and speech features. The hierarchy developed here may utilise sets of features in a variety of combinations. The top level of the hierarchy seeks to capture the main modes of variation of the complete data. However, other levels may be used to model specific relationships between certain features, for example the complete visual head data (modelled as hierarchical appearance model where nodes represent shape and texture of facial features) or the speech data or specific interactions of speech with facial features (e.g. lower face, lips, eyebrows). Again such a model has proven to be robust in tracking facial features and also resynthesising video-realistic new faces.

The principal difficulty in creating such models is in determining which parts should be used, and identifying examples of these parts in the training data.

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2007, LNCS 4418, pp. 225–234, 2007.

The task of finding patterns embedded in the data is a popular research field in computer science [2], [13], [18].

Nonnegative matrix factorization (NMF) [16] is a promising tool in learning the parts of objects and images. NMF imposes non-negativity constraints in its bases and coefficients. These constraints lead to a parts based representation because they allow only additive, not subtractive, combinations. Later in [14] Hoyer presented Sparse non-negative matrix factorisation (sparse NMF) with an adjustable sparseness parameter. This allows it to discover parts-based representations that are qualitatively better than those given by the basic NMF. Because of its parts-based representation property, NMF and its variations have been used to image classification [3], [8], [9], [11], face expression recognition [4], face detection [5], face and object recognition [19], [20], [21].

In all of those papers the number of data parts was quite large and was chosen manually. In this paper, we propose intrinsic dimensionality estimation to find correct number of the parts.

The novelty of this paper is that we use NMF for an automatic data mask construction. We consider the construction of our model in Section 2 and demonstrate the effectiveness of our algorithm applying it to the different data types: talking head data, emotional head data and articulated human motion data in Section 3. Finally, the conclusion is given in Section 4.

2 Partial Data Representation

2.1 Data Format and Preprocessing

Initial data for our algorithm can be represented by a parameterised model or by images. The output generated is a mask identifing different data parts.

This algorithm works best with well aligned data, i.e. the data obtained from a sequence of observations. But it is not really suitable for separation of images of highly articulated objects or objects viewed from significantly different viewpoints into parts.

A normalisation step is needed to make the patterns of interest more evident. Data normalisation is provided as a preprocessing step before NMF, in the same manner as in Li et al. [18].

At the first step of our algorithm we set the number of the data parts. We choose this number to be the same as the intrinsic dimension of the data manifold. We use the k-NN method described in [7] to estimate the intrinsic dimensionality. In this method the dimension is estimated from the length of the minimal spanning tree on the geodesic NN (nearest neighbour) distances computed by the Isomap algorithm [23]. To automate the k-NN method we choose the number of nearest neighbours using the algorithm described in [22].

2.2 Sparse NMF Modification and Initialisation

Classical NMF is a method to obtain a representation of data using non-negativity constraints. These constraints lead to a part-based representation because they only allow additive, not subtractive, combinations of the original data [16]. Given initial data expressed by an $n \times m$ matrix X, where each column is an n-dimensional non-negative vector of the original data (m vectors), it is possible to find two new matrices (W and H) in order to approximate the original matrix:

$$X_{ij} \approx (WH)_{ij} = \Sigma_{l=1}^{r} W_{il} H_{lj} \tag{1}$$

The dimensions of the factorised matrices W and H are $n \times r$ and $r \times m$ respectively. Each column of W contains a basis vector while each column of H contains the weight needed to approximate the corresponding column in X using the bases from W.

Given a data matrix X, the optimal choice of matrices W and H is defined to be those nonnegative matrices that minimise the reconstruction error between X and WH. Various error functions have been proposed [17], the most widely used one is the squared error (Euclidean distance) function

$$E(W, H) = \|X - WH\|^2 = \Sigma_{ij}(X_{ij} - (WH)_{ij})^2 \tag{2}$$

However, the additive parts learned by NMF are not necessarily localised, as was pointed out by Li et al. in [18]. To obtain meaningful partial representation we want to restrict energy of each NMF basis to the most significant components only. Therefore we use sparse NMF [14] which proved to be more appropriate in part-based object decomposition than original NMF.

In sparse NMF the objective (2) is minimised under the constraints that all columns of W and rows of H have common sparseness σ_W and σ_H respectively. The sparseness $\sigma(x)$ is defined by the relation between the Euclidean norm $\|.\|_2$ and 1-norm $\|x\|_1 := \Sigma_i \mid x_i \mid$ as follows

$$\sigma(x) := \frac{\sqrt{n} - \frac{\|x\|_1}{\|x\|_2}}{\sqrt{n} - 1} \tag{3}$$

if $x \in R^n \setminus 0$. Since $\frac{1}{n}\|x\|_1 \leq \|x\|_2 \leq \|x\|_1$ equation (3) is bounded $0 \leq \sigma(x) \leq 1$. In particular, $\sigma(x) = 0$ for minimal sparse vectors with equal non-zero components, and $\sigma(x) = 1$ for maximally sparse vectors with all but one vanishing components.

Sparse NMF Modification: Random Acol Initialisation. It is well known that good initialisation can improve the speed and the accuracy of the solutions of many NMF algorithms [24]. Langville et al. proposed in [15] random Acol initialisation as an inexpensive and effective initialisation technique. Random Acol forms an initialisation of each column of the basis matrix W by averaging p random columns of X. We use the random Acol technique for our modified sparse NMF instead of a random initialisation.

So far, we have three unspecified parameters in our method: initialisation parameter p and sparseness parameters σ_W and σ_H. To automate the algorithm we put p to $[\frac{m}{r}]$ value. We learn useful features from basis W and leave the sparseness of H unconstrained. For all our experiments we set σ_W to 0.78 for simplicity. For more accurate estimation of the sparseness one can use the method described at [12].

Sparse NMF Modification: Earth Mover's Distance. Figure 1 shows the example of sparse NMF basis from the Hoyer paper [14]. It is can be seen that there are significant similarities among the learned bases. Guillamet and Vitria proposed that the Earth mover's distance (EMD) is better suited to this problem because one can explicitly define a distance which will depend on the basis correlation [10].

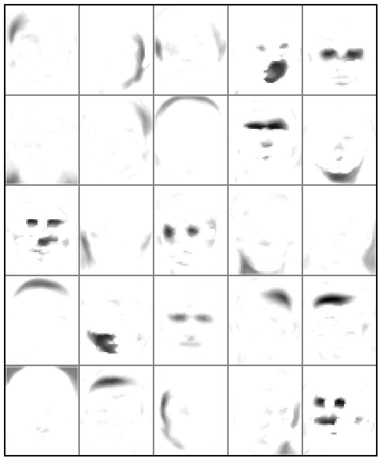

Fig. 1. Features learned from the ORL database using sparse NMF

EMD can be stated as follows: let I be a set of suppliers, J a set of consumers and d_{ij} the cost to ship a unit of supply from $i \in I$ to $j \in J$. We define d_{ij} as the Euclidean distance. We want to find a set of f_{ij} that minimises the overall cost:

$$dist(x, y) = \min \Sigma_{i \in I} \Sigma_{j \in J} d_{ij} f_{ij} \quad (4)$$

subject the following constraints:

$$f_{ij} \geq 0, x_i \geq 0, y_j \geq 0, i \in I, j \in J$$

$$\sum_{i \in I} f_{ij} \leq y_j, j \in J$$

$$\sum_{j \in J} f_{ij} \leq x_i, i \in I$$

$$\Sigma_{i \in I} \Sigma_{j \in J} d_{ij} f_{ij} = min(\Sigma_{i \in I} x_i, \Sigma_{j \in J} y_j)$$

where x_i is the total supply of supplier i and y_j is the total capacity of consumer j.

We used EMD as the distance metric instead the Euclidean distance as in our experiments we obtain better partitioning with this metric.

2.3 Data Postprocessing: Mask Construction

After getting the modified sparse NMF basis we need to analyse the result. At this step we produce a mask of the data by construction of the boundaries between the basis vectors.

We consider the mask construction for the images as it demonstrates the result most clearly. However, our algorithm can be used on a wide range of data. In the next section we describe postprocessing example for 3D human motion data.

Examples of the modified sparse NMF basis are shown in Figures 3 and 6. Each of the basis vectors represents a part of the original image. There is substantial noise in each vector, and some vectors contain several separated parts.

First we consider each vector of the basis separately to reduce the noise. We define a vector element as noise if a 7×7 pixel square centered at this element has any other pixels with zero values. After deleting such components we label each nonzero basis vector component according to its vector number and merge the vectors.

Next we use region growing technique which is a basic yet effective method. Region growing [1] is a technique which begins with a seed location and attempts to merge neighboring pixels until no more pixels can be added to it. Because of basis sparseness, we have considerable amount of pixels that were not assigned a label and now needs one to be allocated. These errors have to be removed in a second postprocessing step. The most dominant regions, i.e. the regions with largest component values, are selected as seed regions for a region growing process. Region growing is implemented as a morphological operation. A 3×3 square is moved over the merged basis. When a neighbor to the point of interest (the center of the square) has a label assigned, the point of interest is checked for compatibility to that region. In case it is found to be compatible (i.e. all point neighbors belongs to the same basis label), it is assigned the label of the corresponding region. If there are conflicting regions, i.e. there are different regions adjacent to the point of interest, the largest region is preferred. This is also the case if the center pixel is already labeled.

When this process is completed, every pixel is assigned one of the possible basis labels, this completes our data partitioning algorithm.

3 Experimental Results

In this section we evaluate how the proposed algorithm processes several real world data sets with different characteristics. The first data considered in Section 3.1 is the talking head. On this example we show how our algorithm works with large images where data variation concentrated mainly on a small region (mouth). Next we consider facial emotional head data. Here we have data variation across the whole image. Section 3.3 describes the model visualisation ability with 3D coordinates of a walking person.

3.1 Talking Head Data

The algorithm described in the previous section was tested on the data from [6]. Initially, it was the video of a speaker reading a text, recorded at 25fps. The subject was recorded front-on with as little out of plane head movement as possible.

We extracted the texture from each frame of the video as described in [6]. Figure 2 shows examples of the texture. We perform the data normalisation [18] to improve algorithm convergence and to make the patterns of interest more evident. Intrinsic dimensionality of the data chosen by the automated k-NN method is eight. Setting the number of basis vectors to 8, we perform the second step of our algorithm. The result of this step is shown at Figure 3. One can see parts of the face there: eyes, cheeks, chin.

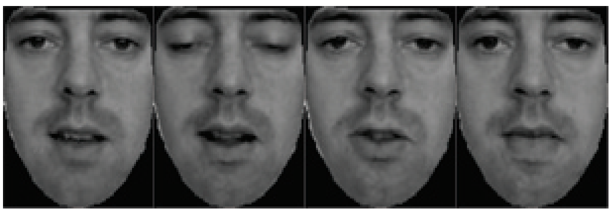

Fig. 2. Talking head data: examples

We use the postprocessing algorithm described in Section 2.3 to automatic basis analysis. Figure 4 shows the mask generated by our algorithm, data mask, which looks appropriate. It can be seen that the automatically constructed partitioning extracts the most important features of the face. We have eyes region, three mouth regions (upper lip, lower lip, inside part of the mouth), cheeks, chin, cheek bones and eyebrow regions. Such partitioning could be very useful for the further data analysis offering us a trade-off between keeping fine detail in the data and the large data dimensionality.

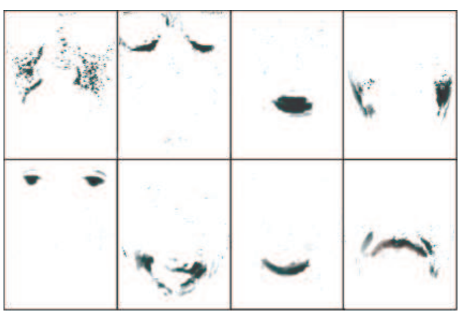

Fig. 3. Talking head data: modified NMF basis

Fig. 4. Talking head data: mask

3.2 Emotion Head Data

For our next experiment we used data sets from two different people. Each person performed a different facial expressions: happiness, sadness and disgust, see Figure 5 for the examples.

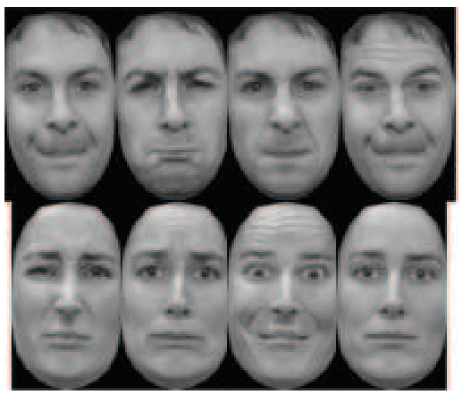

Fig. 5. Emotional head data one (top) and two (bottom)

Unlike the talking head data, the emotional head data has lots of variation across the whole face. In order to see how our algorithm can deal with emotions, we apply it to each data set. As expected, both sets have the same estimated intrinsic dimensionality, equal to 6. Thus we got 6 modified sparse NMF basis vectors which are shown at Figure 6. The basis vectors for the emotion head data look similar to the vectors from the previous example. Because the mouth variation is not significant for this example, a vector representing this variation is missed here, while we have 3 mouth variation vectors for the talking head. Instead we got more vectors to represent other face parts which displayed greater variation in these data sets.

To analyse the modified sparse NMF basis we perform data postprocessing, as described in Section 2.3. The results are shown in Figure 7. Again, the results are natural and similar to the talking head mask, but with more attention paid

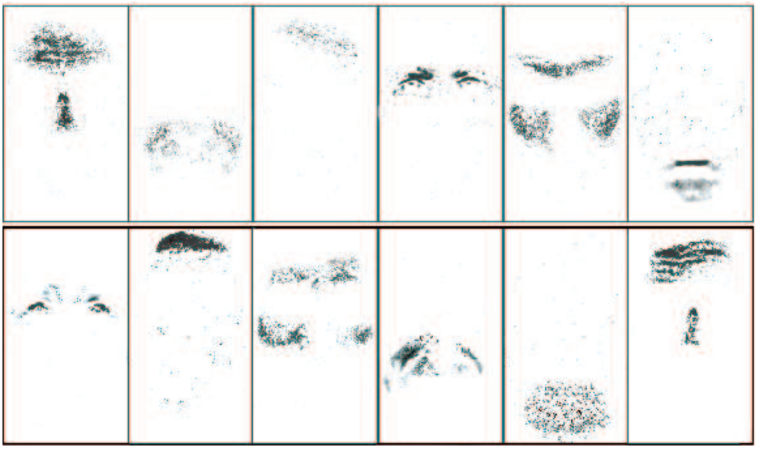

Fig. 6. Emotional data one (top) and two (bottom): modified NMF bases

Fig. 7. Emotional data one (left) and two (right): masks

to the general face details. For example, we got a vector which represents a nose. Our algorithm extracts features which are significant for those particular data.

3.3 Motion Data

We tested our algorithm with two motion data sets. The first set represents the motion of a walking person, consisting of two steps, right turn and one step. The second set represents a two step motion without turn. The initial feature parameters represent the coordinates of human (arms, legs, torso) in the 3D space. Each pose is characterised by 17 points.

Following the preprocessing step of our algorithm, we choose the intrinsic dimensionality [7], which is 2 for each set. After running modified NMF we get a sparse basis sets for analysis.

We cannot apply the postprocessing step from Section 2.3 because of the data type. Therefore we perform postprocessing in a different way.

We define that a 3D pose junction point belongs to the basis in which this point has maximum summed basis coefficients. Figure 8 illustrates the bases for both sets. On the left hand side of Figure 8 one can see data partitioning for the walking with turn. The first basis vector here is represented by the torso and forearms, and the second one is represented by legs and shoulders. On the right hand side of Figure 8 we show data partitioning for the straight walking. Here we have a different partitioning, which consists of two pieces too. The first basis vector represents a motion of the right arm and the left leg, while the second basis vector represents the left arm, the torso and the right leg. Such partitioning isolates variation in subregions from the rest of the body and provides a high degree of control over different body parts. For example, it is straight forward to find out which part is responsible for moving the legs (the first case), or describes relationships between legs and arms movements (the second case).

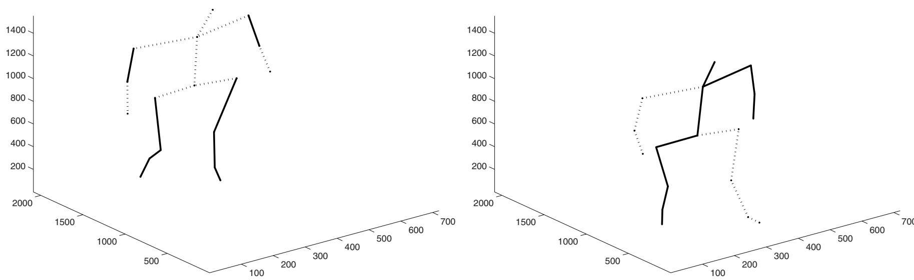

Fig. 8. 3D motion data partitioning - different bases are represented by different line drawing styles

4 Conclusions

We have described a new algorithm for automatic data decomposition using a modified sparse NMF basis analysis, in which the number of basis vectors is selected to be the same as the estimated intrinsic dimensionality of the data. Segmentation is then performed by applying region growing to the set of basis vectors.

We demonstrate the algorithm's ability to produce good partitioning of real data sets. In these examples we show that our algorithm extracts the most important features for the particular data set. Such partitioning provides a powerful tool for automating construction of parts based data models.

In future work we hope to extend our algorithm for more complex cases, such as highly articulated objects. We also plan to improve the postprocessing step by performing more advanced segmentation.

References

1. R. Adams and L. Bischof. Seeded region growing. In *Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 641–647, 1994.
2. M. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182, 1999.
3. G. Buchsbaum and O. Bloch. Color categories revealed by non-negative matrix factorization of munsell color spectra. *Vision Research*, 42:559–563, 2002.

4. I. Buciu and I. Pitas. Application of non-negative and local non-negative matrix factorization to facial expression recognition. In *Proc. of ICPR*, volume 1, pages 288–291, 2004.
5. X. Chen, L. Gu, S.Z. Li, and H.J. Zhang. Learning representative local features for face detection. In *Proc. of ICPR*, volume 1, pages 1126–1131, 2001.
6. D.P. Cosker, A.D. Marshall, P.L. Rosin, and Y. Hicks. Speech driven facial animation using a hierarchical model. In *Proc. of IEE Vision, Image and Signal Processing*, volume 151, pages 314–321, 2004.
7. J. Costa and A. O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. In *Proc. of IEEE Trans. on Signal Processing*, volume 52, pages 2210–2221, 2004.
8. D. Guillamet, M. Bressan, and J. Vitrià. A weighted non-negative matrix factorization for local representations. In *Proc. of CVPR*, volume 1, pages 942–947, 2001.
9. D. Guillamet and J. Vitrià. Non-negative matrix factorization for face recognition. In *Proc. of CCIA*, pages 336–344, 2002.
10. D. Guillamet and J. Vitrià. Evaluation of distance metrics for recognition based on non-negative matrix factorization. *Pattern Rocognition Letters*, 24(9-10):1599–1605, 2003.
11. D. Guillamet, J. Vitriá, and B. Schiele. Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters*, 24(14):2447–2454, 2003.
12. M. Heiler and C. Schnorr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *JMLR*, (7):1385–1407, 2006.
13. G. Hinton, Z. Ghahramani, and Y. Teh. Learning to parse images. In *Proc. of NIPS*, pages 463–469, 1999.
14. P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.
15. A. N. Langville, C. D. Meyer, R. Albright, J. Cox, and D. Duling. Initializations for the nonnegative matrix factorization. In *Proc. of the 12 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.
16. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
17. D.D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of NIPS*, pages 556–562, 2000.
18. S. Li, X. Hou, and H. Zhang. Learning spatially localized, parts-based representation. In *Proc. of IEEE CVPR*, volume 1, pages 207–212, 2001.
19. W. Liu and N. Zheng. Learning sparse features for classification by mixture models. *Pattern Recognition Letters*, 25(2):155–161, 2004.
20. W. Liu and N. Zheng. Non-negative matrix factorization based methods for object recogni-tion. *Pattern Recognition Letters*, 25:893–897, 2004.
21. M. Rajapakse, J. Tan, and J. Rajapakse. Color channel encoding with nmf for face recognition. In *Proc. of ICIP*, volume 3, pages 2007–2010, 2004.
22. O. Samko, A.D. Marshall, and P.L. Rosin. Selection of the optimal parameter value for the isomap algorithm. *Pattern Rocognition Letters*, 27:968–979, 2006.
23. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.
24. S. Wild. Seeding non-negative matrix factorizations with the spherical k-means clustering. *Master's Thesis, University of Colorado*, 2003.