Basis Decomposition of Motion Trajectories Using Spatio-temporal NMF

Sven Hellbach, Julian Eggert, Edgar Körner, Horst-Michael Groß

2009

Preprint:

This is an accepted article published in Artificial Neural Networks - {ICANN} 2009, 19th International Conference. The final authenticated version is available online at: https://doi.org/[DOI not available]

Basis Decomposition of Motion Trajectories using Spatio-Temporal NMF

Sven Hellbach[1], Julian P. Eggert[2], Edgar Körner[2], and Horst-Michael Gross[1]

[1] Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Labs, POB 10 05 65, 98684 Ilmenau, Germany
sven.hellbach@tu-ilmenau.de
[2] Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany
julian.eggert@honda-ri.de

Abstract. This paper's intention is to present a new approach for decomposing motion trajectories. The proposed algorithm is based on non-negative matrix factorization, which is applied to a grid like representation of the trajectories. From a set of training samples a number of basis primitives is generated. These basis primitives are applied to reconstruct an observed trajectory. Hence, the reconstruction information can be used for classification. Furthermore, the basis primitives can be used to predict the observed movement. For the experiments, real movement data is used to evaluate several aspects of the proposed approach. In particular, the focus of the experiments is set to the prediction task.

Key words: Non-negative Matrix Factorization, Prediction, Movement Data, Robot, Motion Trajectories

1 Introduction

The understanding and interpretation of trajectories is a crucial component in dynamic visual scenes with multiple moving items. Nevertheless, this problem has been approached very sparsely by the research community. Most approaches for describing motion patterns, like [1], rely on a kinematic model for the observed human motion. This causes the drawback, that those approaches are difficult to adapt to other objects. Here, we aim at a generic, model-independent framework for decomposition, classification and prediction. In this paper we focus on the decomposition and prediction problem, while the classification is not yet further investigated.

Consider the simple task for a robot of grasping an object which is handed over by the human interaction partner. To avoid a purely reactive behaviour, which might lead to 'mechanical' movements of the robots, it is necessary to predict the further movement of the human's hand.

In [2] an interesting concept for a decomposition task is presented. Like playing a piano a basis alphabet – the different notes – are superimposed to reconstruct the observation (the piece of music). The much less dimensional description of when each basis primitive is used, can be used for further processing.

While the so called piano model relies on a set of given basis primitives, our approach is able to learn these primitives from the training data.

Beside the standard source separation approaches, like PCA and ICA, another promising algorithm exists. It is called non-negative matrix factorization (NMF) [3]. The system of basis vectors which is generated by the NMF is not orthogonal. This is very useful for motion trajectories, since a basis primitive is allowed to share a common part of trajectory with other primitives and to specialize later.

The next section introduces the standard non-negative matrix factorization approach and two extension that can be found in the literature. In section 3 the new approach for decomposing motion trajectories is presented. The experiments with their conditions and results are presented in section 4, while the paper concludes in section 5.

2 Non-negative Matrix Factorization

Like other approaches, e. g. PCA and ICA, non-negative matrix factorization (NMF) [3] is meant to solve the source separation problem. Hence, a set of training data is decomposed into basis primitives:

$$\mathbf{V} \approx \mathbf{W} \cdot \mathbf{H} \tag{1}$$

Each training data sample is represented as a column vector $\mathbf{V}_i$ within matrix $\mathbf{V}$. Each column of matrix $\mathbf{W}$ stands for one of the basis primitives. In matrix $\mathbf{H}$ the element H_i^j determines how the basis primitive $\mathbf{W}_j$ is activated to reconstruct training sample $\mathbf{V}_i$. Since NMF is an iterative approach, the training data $\mathbf{V}$ can only be approximated by the product of $\mathbf{W}$ and $\mathbf{H}$. This product will be referred to as reconstruction $\mathbf{R} = \mathbf{W} \cdot \mathbf{H}$ later.

Unlike PCA or ICA, NMF aims to a decomposition, which only consists of non-negative elements. This means that the basis primitives can only be accumulated. There exists no primitive which is able to erase a 'wrong' superposition of other primitives. This leads to a more specific set of basis primitives, which is a clear advantage for certain applications, like face recognition [4].

For generating the decomposition, optimization based methods are used. Hence, an energy function E has to be defined:

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \left\| \mathbf{V} - \mathbf{W} \cdot \mathbf{H} \right\|^2 \tag{2}$$

By minimizing the energy equation, it is now possible to achieve a reconstruction using the matrices $\mathbf{W}$ and $\mathbf{H}$. This reconstruction is aimed to be as close as possible to the training data $\mathbf{V}$. No further constraints are given in the standard formulation of the NMF. As it can be seen in equation 2, the energy function depends on the two unknown matrices $\mathbf{W}$ and $\mathbf{H}$.

Since, both matrices usually have a large number of elements, the optimization problem seems to be an extensive task. Fortunately, each training sample

can be regarded independent from the others:

$$\mathbf{V_i} \approx \sum_j H_i^j \cdot \mathbf{W_j} \tag{3}$$

Furthermore, both matrices are adapted in an alternating fashion. This helps to reduce the number of dimension for the optimization process and allows a training with fewer examples. The algorithm is depicted in the following description:

1. Calculate the reconstruction

$$\mathbf{R}_i = \sum_j H_i^j \mathbf{W}_j \tag{4}$$

2. Update the activities

$$H_i^j \leftarrow H_i^j \odot \frac{\mathbf{V}_i^T \mathbf{W}_j}{\mathbf{R}_i^T \mathbf{W}_j} \tag{5}$$

3. Calculate the reconstruction with the new activities

$$\mathbf{R}_i = \sum_j H_i^j \mathbf{W}_j \tag{6}$$

4. Update the basis vectors

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i H_i^j \mathbf{V}_i}{\sum_i H_i^j \mathbf{R}_i} \tag{7}$$

Steps 1 to 4 are iterated until a defined convergence criteria is reached. For the criteria the energy function can be used in a usual fashion. Where the operation $\odot$ denotes a component-wise multiplication. Beside this analytical description non-negative matrix factorization can also be formulated with connectionist methods. This is shown in [5], for example.

2.1 Sparse Coding

As it could be seen in equation 2 the energy function is formulated in a very simple way. This results in a decomposition, which is quite arbitrary with no further characteristics. This can lead, for example, to redundant information. Especially, if the number of basis primitives is chosen higher than needed to decompose the given training data. To compensate this drawback, it is useful to introduce a constraint which demands a sparse activation matrix, like it was introduced in [6]. This avoids the fact, that several basis primitives are activated at the same time, and hence are being superimposed.

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \left\| \mathbf{V} - \mathbf{W} \cdot \mathbf{H} \right\|^2 + \lambda \sum_{i,j} H_i^j \tag{8}$$

The influence of the sparsity constraint can be controlled using parameter λ. In this paper, we only discuss a special case for the sparsity term. A more detailed discussion can be found in [6]. The algorithmic description is similar to the one of the standard NMF. The only thing that has to be considered is that the basis primitives need to be normalized.

2.2 Transformation Invariance

Beside the sparsety constraint an other extension to NMF has been published in [7]. The concept of transformation invariance allows to move, rotate and scale the basis primitves reconstructing the inpur. In this way, each possible transformation doesn't need to be covered by the basis primitives, but gets handled by the algorithm. This is achieved by adding a transformation matrix $\mathbf{T}$ to the decomposition formulation:

$$\mathbf{V} \approx \mathbf{T} \cdot \mathbf{W} \cdot \mathbf{H} \tag{9}$$

However the activation matrix $\mathbf{H}$ has to be adapted in a way that each possible transformation carries its own activation. Hence, the matrix $\mathbf{H}$ becomes an activation tensor $\mathbf{H^m}$, while $\mathbf{m}$ is a vector describing the transformation parameters (rotation, scaling and translation).

$$\mathbf{V_i} \approx \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \cdot \mathbf{T^m} \cdot \mathbf{W_j} \tag{10}$$

For each allowed transformation the corresponding activity has to trained individually.

3 Decomposing Motion Trajectories

For being able to decompose and to predict the trajectories of the surrounding dynamic objects, it is necessary to identify them and to follow their movements. For simplification, a tracker is assumed, which is able to provide such trajectories in real-time. A possible tracker to be used is presented in [8]. The given trajectory of the motion is now interpreted as a time series $\mathcal{T}$ with values $\boldsymbol{s}_i$ for time steps $i = 0, 1, \ldots, n-1$:

$$\mathcal{T} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1}) \tag{11}$$

It could now be possible to present the vector $\mathcal{T}$ directly to the NMF approach. But this could result in an unwanted behavior, while trying to reconstruct the motion by use of the basis primitives. Imagine, two basis primitives, one representing a left turn and another representing a right turn. A superposition of those basis primitives would result in a straight movement.

The goal is to have a set of basis primitives, which can be concatenated one after the other. Furthermore, it is necessary for a prediction task to be able to formulate multiple hypotheses. For achieving these goals, the x-t-trajectory is transferred into a grid representation, as it is shown in figure 1. Then, each grid cell (x_i, t_j) represents a certain state (spatial coordinate) x_i at a certain time

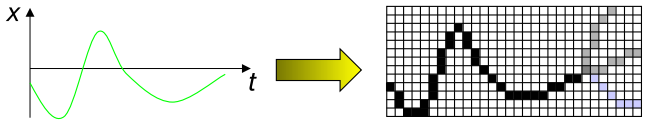


Fig. 1. Motion Trajectories are transferred into a grid representation. A grid cell is set to 1 if it is in the path of the trajectory and set to zero otherwise. Each dimension has to be regarded separately. During the prediction phase multiple hypotheses can be gained by superimposing several basis primitives. This is indicated with the gray trajectories on the right side of the grid.

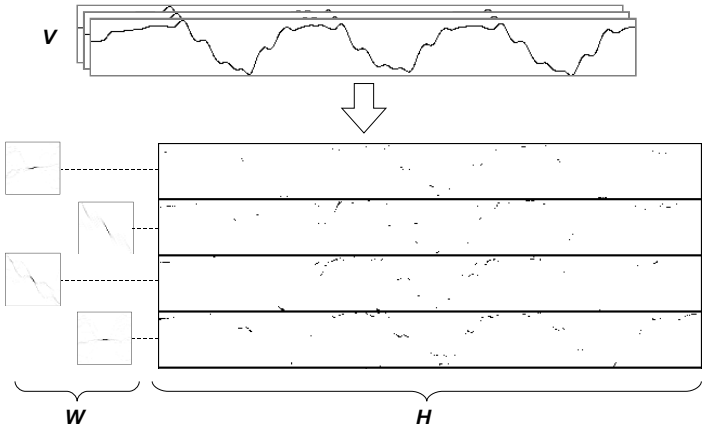


Fig. 2. Training with Spatio-Temporal NMF. Given is a set of training samples in matrix **V**. The described algorithm computes the weights **W** and the corresponding activities **H**. Only the weights are used as basis primitives for further processing.

t_j. Since most of the state-of-the-art navigation techniques rely on grid maps [9], the prediction can be integrated easily. This 2D-grid is now presented as image-like input to the NMF algorithm using the sparsity constraint as well as transformation invariance (See section 2.1 and 2.2 respectively). Using the grid representation of the trajectory, also supports the non-negative character of the basis components and its activities.

While applying an algorithm for basis decomposition to motion trajectories it seems to be clear that the motion primitives can undergo certain transformations to be combined to the whole trajectory. For example, the same basis primitive standing for a straight move can be concatenated with an other one standing for a left turn. Hence, the turning left primitive has to be moved to the end of the straight line. Hence, transformation invariance is needed while decomposing motion data. For our purposes, we concentrate on translation. This makes it possible to simplify the calculations and to achieve real time performance.

The sparse coding constraint helps to avoid trivial solutions. Since the input can be compared with a binary image, one possible solution would be a basis

1. Normalize the basis vectors according to

$$\overline{\mathbf{W}}_j = \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|} \tag{12}$$

2. Calculate the reconstruction

$$\mathbf{R}_i = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{T}^{\mathbf{m}} \overline{\mathbf{W}}_j \tag{13}$$

3. Update the activities

$$H_i^{j,\mathbf{m}} \leftarrow H_i^{j,\mathbf{m}} \odot \frac{\mathbf{V}_i^T \mathbf{T}^{\mathbf{m}} \overline{\mathbf{W}}_j}{\mathbf{R}_i^T \mathbf{T}^{\mathbf{m}} \overline{\mathbf{W}}_j} \tag{14}$$

4. Calculate the reconstruction with the new activities

$$\mathbf{R}_i = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{T}^{\mathbf{m}} \overline{\mathbf{W}}_j \tag{15}$$

5. Update the basis vectors

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{V}_i^T \mathbf{T}^{\mathbf{m}} + \overline{\mathbf{W}}_j \overline{\mathbf{W}}_j^T \sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{R}_i^T \mathbf{T}^{\mathbf{m}}}{\sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{R}_i^T \mathbf{T}^{\mathbf{m}} + \overline{\mathbf{W}}_j \overline{\mathbf{W}}_j^T \sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{V}_i^T \mathbf{T}^{\mathbf{m}}} \tag{16}$$

Fig. 3. Algorithmic description of the Spatio-temporal NMF.

component with single grid cell filled. The trajectory is then simply copied into the activities.

3.1 Training Phase

The goal of the training phase is to gain a set of basis primitives which allow to decompose an observed and yet unknown trajectory (see Fig. 2). As it is discussed in section 3, the training samples are transferred into a grid representation. These grid representations are taken as input for the NMF approach and are therefor represented in matrix $\mathbf{V}$. On this matrix $\mathbf{V}$ the standard NMF approach, extended by sparsity constraint and by translation invariance, is applied. The algorithm is depicted in Fig. 3.

Beside the computed basis primitives, the NMF algorithm also provides the information how each of the training samples can be decomposed by these basis primitives. This information is not used in the application phase.

3.2 Application Phase

As it is indicated in Fig. 4, from the training phase a set of motion primitives is given. During the application phase, we assume that the motion of a dynamic object (e. g. a person) is tracked continuously. For getting the input for the NMF algorithm, a sliding window approach is taken. A certain frame in time

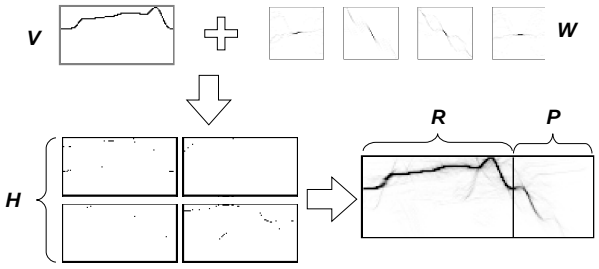


Fig. 4. The basis primitives **W**, which were computed during the training, are used to reconstruct (matrix **R**) the observed trajectory **V**. This results in a set of activities – one for each basis primitive – which describe on which position in space and time a certain primitive is used. Beside the reconstruction of the observed trajectory (shown in Fig. 4), it is furthermore possible to predict a number of time steps into the future. Hence, the matrix **R** is extended by the prediction horizon **P**.

is transferred into the already discussed grid like representation. For this grid the activation of the basis primitives is determined by trying to reconstruct the input. For the computation the algorithm is identical to the one depicted in Fig. 3 beside that step 2 and 3 can be skipped.

The standard approach to NMF implies that each new observation at the next time step demands a new random initialization for the optimization problem. Since, an increasing column number in the grid representation stand for an increase in time, the trajectory is shifted to the left while moving further in time. Because the solution of the NMF is unambiguous, the shift can also be observed in the activities after the convergence. To reduce the number of iterations until convergence, the shifted activities from the previous time step are used as initialization for the current one.

To fulfill the main goal discussed in this paper – the prediction of the observed trajectory into the future – the proposed algorithm has to be extended. Since, the algorithm contains the transformation invariance constraint, the computed basis primitives can be translated to an arbitrary position on the grid. This means they can also be moved in a way that they exceed the borders of the grid and are therefor clipped. Up to now, the size of reconstruction was chosen to be the same size as the input grid. To be able to solve the prediction task, we simply extend the reconstruction grid to the right – or into the future (see Fig. 4). So the previously clipped information is available for prediction.

4 Evaluation

Taking a closer look at the introducing example scenario reveals that a robust identification and tracking of the single body parts is needed. To be compareable and to avoid errors from the tracking system influencing the test results movement data from the Perception Action Cognition Lab at the University of Glasgow [10] is used. A number of 30 persons is recorded performing different actions in different moods. The movement data has a resolution of 60 time steps

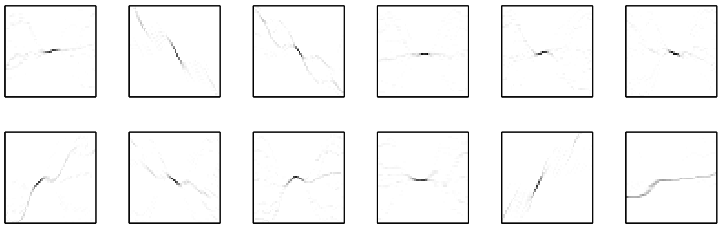


Fig. 5. Basis primitives gained by Spatio-Temporal NMF. The value for each grid cell is coded in gray scale from white (low) to black (high). A certain value stands for the influence of this grid cell. Hence, light gray parts can be superimposed well, while dark gray to black parts tend to be unambiguous.

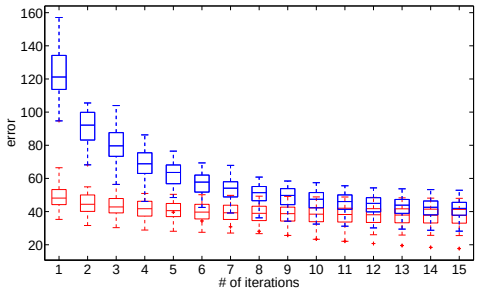


Fig. 6. Box whiskers plot showing the characteristics of the energy function (see eqn. 2 for 15 iteration steps. For the upper (blue) plot the activities are initialized randomly after each shift of the input data. For the lower (red) curve the activities from the previous computations are shifted and used as initialization.

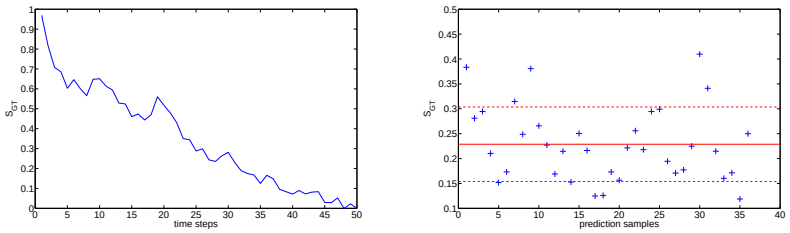
per second, so that an average prediction of about 50 steps means a prediction of 0.83 seconds into the future. Since most trackers work with a lower resolution, a prediction further into the future is still possible.

In the next sections, two aspects of the proposed algorithm is investigated in detail. First, it is shown that activity shifting brings a great benefit towards real time performance. Afterwards the focus is set to the quality of the prediction part.

For the experiments, the size of the basis primitives was chosen to be 50×50 grid cells (for an example see Fig. 5). The input grid size during the training phase was set to 500×50 for each of the trajectories and to 100×50 during application phase.

4.1 Activity shifting

In section 3.2 is has been discussed, that the information from the previous time step can be used as initialization for the current. Figure 6 shows the energy function, which is defined in equation 2 for both ways of initialization. It is



(a) (b)

Fig. 7. (a) The mean correlation S_{GT} (see eqn. 17) between the ground truth trajectory and the prediction is plotted for each time step of the prediction horizon. A fit value of 1.0 stands for a perfect prediction over the whole prediction horizon. As it is expected the accuracy of the prediction decreases for a longer prediction period. (b) The plot shows the prediction accuracy for predictions along a sample trajectory. The 36 predictions were performed at each tenth time step of the chosen trajectory. A fit value of 1.0 stands for a perfect prediction over the whole prediction horizon. The constant and dotted lines (red) indicate mean and variance respectively.

plotted only for a number of iteration steps, since the effect can be observed early. For the upper (blue) plot the initialization is done completely randomly. For the lower (red) curve the activities from the previous computations are shifted and used as initialization. It can clearly be seen that the convergence is faster by a number of about 10 steps in average.

4.2 Prediction

For evaluating the quality of the prediction, the prediction is compared with grid representation of the actual trajectory $\mathbf{G}$. For each occupied grid cell the value of the normalized prediction is added. The sum is divided by the length of the trajectory:

$$S_{GT} = \frac{1}{|T|} \sum_{t \in T} \frac{\mathbf{P}_t^T \cdot \mathbf{G}_t}{\sum_i \mathbf{G}_t^i} \tag{17}$$

The normalization of the prediction is done separately for each time slice (column in the grid).

The basis primitives can at most be shifted by their width out of the reconstruction grid $\mathbf{R}$. So the theoretically maximum size of the prediction horizon equals the width of the basis primitives. Practically this maximum can not be reached, because the basis primitives need a reliable basis in the part where the input is known. Nevertheless, we have chosen to use the theoretical maximum as basis for the evaluation.

The results are depicted in Fig 7. The first plot (Fig. 7(a)) shows the expected decrease of the average prediction quality over the prediction horizon. Nevertheless, the decrease is not to steep and no sudden collapses can be observed. For Fig. 7(b) an example trajectory has been selected for the reasons of clearness. The plot is intended to show how the algorithm behaves in practical applications. The predictions were performed at each tenth time step of

the chosen trajectory. A fit value of 1.0 stands for a perfect prediction over the maximum prediction horizon, with only a single hypothesis for the prediction. The value decreases significantly with multiple hypotheses being present.

5 Conclusion and Outlook

This paper presented a new approach for decomposing motion trajectories using non-negative matrix factorization. To solve this problem, sparsity constraint and transformation invariance have been combined. The trajectories were then presented in a grid representation. It could be demonstrated, that the concept of activity shifting clearly decreases the number of iterations needed until convergence. Furthermore it could be seen, that the proposed algorithm is able to predict the motion into the future. At the moment, the prediction is only formulated to get a possibilistic information. It would be interesting to extend the concept, to provide a probabilistic description. It was mentioned that is possible to use the resulting activities as input for a classifier. Further tests are needed to evaluate the feasibility of this idea.

References

1. Hoffman, H., Schaal, S.: A computational model of human trajectory planning based on convergent flow fields. In: Abstracts of the 37st Meeting of the Society of Neuroscience. (2007)
2. Cemgil, A., Kappen, B., Barber, D.: A generative model for music transcription. IEEE Transactions on Speech and Audio Processing **14** (2006) 679694
3. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing **13** (2001) 556–562
4. Rajapakse, M., Wyse, L.: NMF vs ICA for face recognition. In: ISPA 2003. Volume 2. (2003) 605–610
5. Rebhan, S., Sharif, W., Eggert, J.: Incremental Learning in the Non-negative Matrix Factorization. In: ICONIP 2008, Springer (Forthcoming)
6. Eggert, J., Körner, E.: Sparse Coding and NMF. In: IJCNN. (2004) 2529 – 2533
7. Eggert, J., Wersing, H., Körner, E.: Transformation-invariant represenatation and NMF. In: IJCNN. (2004) 2535 – 2539
8. Scheidig, A., Müller, S., Martin, C., Gross, H.M.: Generating person's movement trajectories on a mobile robot. In: Proc. of RO-MAN. (2006) 747 – 752
9. Elfes, A.: Using Occupancy Grids for Mobile Robot Perception and Navigation. Computer **12**(6) (June 1989) 46–57
10. http://paco.psy.gla.ac.uk/data_ptd.php