Smooth Component Analysis as Ensemble Method for Prediction Improvement

Ryszard Szupiluk[1,2], Piotr Wojewnik[1,2], and Tomasz Ząbkowski[1,3]

[1] Polska Telefonia Cyfrowa Ltd., Al. Jerozolimskie 181, 02-222 Warsaw, Poland
{rszupiluk,pwojewnik,tzabkowski}@era.pl
[2] Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland
[3] Warsaw Agricultural University, Nowoursynowska 159, 02-787 Warsaw, Poland

Abstract. In this paper we apply a novel smooth component analysis algorithm as ensemble method for prediction improvement. When many prediction models are tested we can treat their results as multivariate variable with the latent components having constructive or destructive impact on prediction results. We show that elimination of those destructive components and proper mixing of those constructive can improve the final prediction results. The validity and high performance of our concept is presented on the problem of energy load prediction.

1 Introduction

The blind signal separation methods have applications in telecommunication, medicine, economics and engineering. Starting from separation problems, BSS methods are used in filtration, segmentation and data decomposition tasks [5,11]. In this paper we apply the BSS method for prediction improvement in case when many models are tested.

The prediction problem as other regression tasks aims at finding dependency between input data and target. This dependency is represented by a specific model e.g. neural networks [7,13]. In fact, in many problems we can find different acceptable models where the ensemble methods can be used to improve final results [7]. Usually solutions propose the combination of a few models by mixing their results or parameters [1,8,18]. In this paper we propose an alternative concept based on the assumption that prediction results contain the latent destructive and constructive components common to all the model results [16]. The elimination of the destructive ones should improve the final results. To find the latent components we apply blind signal separation methods with a new algorithm for smooth component analysis (SmCA) which is addressed for signals with temporal structure [4]. The full methodology will be tested in load prediction task [11].

2 Prediction Results Improvement

We assume that after the learning process each prediction result includes two types of latent components: constructive, associated with the target, and destructive, associated

M.E. Davies et al. (Eds.): ICA 2007, LNCS 4666, pp. 277–284, 2007.

with the inaccurate learning data, individual properties of models, missing data, not precise parameter estimation, distribution assumptions etc. Let us assume there is m models. We collect the results of particular model in column vector $\mathbf{x}_i$, i=1,…, m, and treat such vectors as multivariate variable $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_m]^T$, $\mathbf{X} \in R^{m \times N}$, where N means the number of observations. We describe the set of latent components as $\mathbf{S} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, ..., \hat{\mathbf{s}}_k, \mathbf{s}_{k+1}, \mathbf{s}_n]^T$, $\mathbf{S} \in R^{n \times N}$, where $\hat{\mathbf{s}}_j$ denotes constructive component and $\mathbf{s}_i$ is destructive one [3]. For simplicity of further considerations we assume $m = n$. Next we assume the relation between observed prediction results and latent components as linear transformation

$$\mathbf{X} = \mathbf{AS}, \tag{1}$$

where matrix $\mathbf{A} \in R^{n \times n}$ represents the mixing system. The (1) means matrix $\mathbf{X}$ decomposition by latent components matrix $\mathbf{S}$ and mixing matrix $\mathbf{A}$.

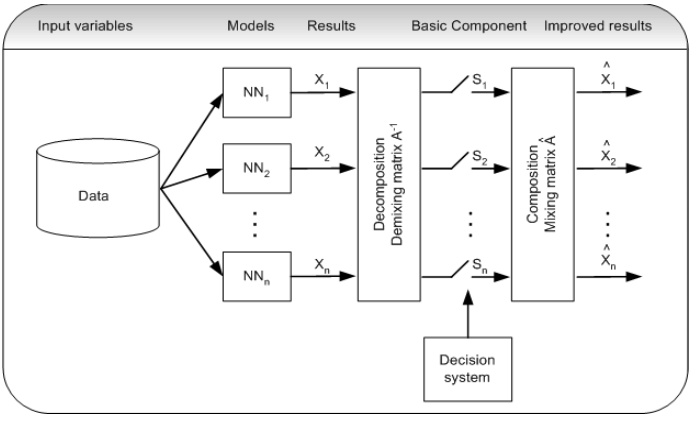

Fig. 1. The scheme of modelling improvement method by multivariate decomposition

Our aim is to find the latent components and reject the destructive ones (replace them with zero). Next we mix the constructive components back to obtain improved prediction results as

$$\hat{\mathbf{X}} = \mathbf{A}\hat{\mathbf{S}} = \mathbf{A}[\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, ..., \hat{\mathbf{s}}_k, \mathbf{0}_{k+1}, ..., \mathbf{0}_n]^T. \tag{2}$$

The replacement of destructive signal by zero is equivalent to putting zero in the corresponding column of $\mathbf{A}$. If we express the mixing matrix as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2 \ldots \mathbf{a}_n]$ the purified results can be described as

$$\hat{\mathbf{X}} = \hat{\mathbf{A}}\mathbf{S} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_k, \mathbf{0}_{k+1}, ..., \mathbf{0}_n]\mathbf{S}, \tag{3}$$

Where $\hat{\mathbf{A}} = [\mathbf{a}_1, \mathbf{a}_2 \ldots \mathbf{a}_p, \mathbf{0}_{p+1}, \mathbf{0}_{p+2} \ldots \mathbf{0}_n]$. The crucial point of the above concept is proper $\mathbf{A}$ and $\mathbf{S}$ estimation. It is difficult task because we have not information which decomposition is most adequate. Therefore we must test various transformations giving us components of different properties. The most adequate methods to solve the first problem seem to be the blind signal separation (BSS) techniques.

3 Blind Signal Separation and Decomposition Algorithms

Blind signals separation (BSS) methods aim at identification of the unknown signals mixed in the unknown system [2,4,10,15]. There are many different methods and algorithms used in BSS task. They explore different properties of data like: independence [2,10], decorrelation [3,4], sparsity [5,19], smoothness [4], non-negativity [12] etc. In our case, we are not looking for specific real signals but rather for interesting analytical data representation of the form (1). To find the latent variables **A** and **S** we can use a transformation defined by separation matrix $\mathbf{W} \in R^{n \times n}$, such that

$$\mathbf{Y} = \mathbf{WX}. \tag{4}$$

where **Y** is related to **S**. We also assume that **Y** satisfies the following relation

$$\mathbf{Y} = \mathbf{PDS}, \tag{5}$$

where **P** is a permutation matrix and **D** is a diagonal matrix [4,10]. The relation (5) means that estimated signals can be rescaled and reordered in comparison to the original sources. These properties are not crucial in our case, therefore **Y** can be treated directly as estimated version of sources **S**. There are some additional assumptions depending on particular BSS method. We focus on methods based on decorrelation, independent component analysis and smooth component analysis.

Decorrelation is one of the most popular statistical procedures for the elimination of the linear statistical dependencies in the data. It can be performed by diagonalization of the correlation matrix $\mathbf{R}_{xx}=E\{\mathbf{X}\mathbf{X}^T\}$. It means that matrix **W** should satisfy the following relation

$$\mathbf{R}_{yy} = \mathbf{W}\mathbf{R}_{xx}\mathbf{W}^T = \mathbf{E}, \tag{6}$$

where **E** is any diagonal matrix. There are many methods utilizing different matrix factorisation leading to the decorrelation matrix **W**, Table 1 [6,17]. The decorrelation is not effective separation method and it is used typically as preprocessing, in general. However, we find it very useful for our analytical representation.

Table 1. Methods of decorrelation possible for models decomposition

Method	Form correlation	Cholesky	EIG (PCA)
Factorisation	$\mathbf{R}_{xx}=\mathbf{R}_{xx}^{1/2}\mathbf{R}_{xx}^{1/2}$	$\mathbf{R}_{xx}=\mathbf{G}^T\mathbf{G}$	$\mathbf{R}_{xx}=\mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$
Decorrelation	$\mathbf{W}=\mathbf{R}_{xx}^{-1/2}$	$\mathbf{W}=\mathbf{G}^{-T}$	$\mathbf{W}=\mathbf{U}^T$

Independent component analysis, ICA, is a statistical tool, which allows decomposition of observed variable **X** into independent components $\mathbf{Y} = [\mathbf{y}_1,\mathbf{y}_2 \ldots \mathbf{y}_n]^T$ [2,4,10]. Typical algorithms for ICA explore higher order statistical dependencies in a dataset, so after ICA decomposition we have got signals (variables) without any linear and non-linear statistical dependencies. To obtain independent components we explore the fact that the joint probability of independent variables can be factorized by the product of the marginal probabilities

$$\overbrace{p_1(\mathbf{y}_1)p_2(\mathbf{y}_2)\dots p_n(\mathbf{y}_n)}^{q_{\mathbf{y}}(\mathbf{Y})} = \overbrace{p_{1\dots n}(\mathbf{y}_1,\mathbf{y}_2,\dots,\mathbf{y}_n)}^{p_{\mathbf{y}}(\mathbf{Y})}. \tag{7}$$

One of the most popular method to obtain (8) is to find such $\mathbf{W}$ that minimizes the Kullback-Leibler divergence between $p_{\mathbf{y}}(\mathbf{Y})$ and $q_{\mathbf{y}}(\mathbf{Y})$ [5]

$$\mathbf{W}_{opt} = \min_{\mathbf{W}} D_{KL}(p_{\mathbf{y}}(\mathbf{WX}) \,\|\, q_{\mathbf{y}}(\mathbf{WX})) = \min_{\mathbf{W}} \int_{-\infty}^{+\infty} p_{\mathbf{y}}(\mathbf{Y}) \log \frac{p_{\mathbf{y}}(\mathbf{Y})}{q_{\mathbf{y}}(\mathbf{Y})} d\mathbf{Y}. \tag{8}$$

There are many numerical algorithms estimating independent components like Natural Gradient, FOBI, JADE or FASTICA [2,4,10].

Smooth Component Analysis, SmCA, is a method of the smooth components finding in a multivariate variable [4]. The analysis of signal smoothness is strongly associated with the definitions and assumptions about such characteristics [9,17]. For signals with temporal structure we propose a new smoothness measure

$$P(\mathbf{y}) = \frac{\frac{1}{N}\sum_{k=2}^{N} |\,\mathbf{y}(k) - \mathbf{y}(k-1)\,|}{\max(\mathbf{y}) - \min(\mathbf{y}) + \delta(\max(\mathbf{y}) - \min(\mathbf{y}))}, \tag{9}$$

where symbol $\delta(.)$ means Kronecker delta, and $P(\mathbf{y}) \in [0,1]$. Measure (9) has simple interpretation: it is maximal when the changes in each step are equal to range (maximal change), and is minimal when data are constant. The Kronecker delta term is introduced to avoid dividing by zero. The range calculated in denominator is sensitive to local data, what can be avoided using extremal values distributions.

The components are taken as linear combination of signals $\mathbf{x}_i$ and should be as smooth as possible. Our aim is to find such $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2 \dots \mathbf{w}_n]$ that for $\mathbf{Y} = \mathbf{WX}$ we obtain $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2 \dots \mathbf{y}_n]^T$ where $\mathbf{y}_1$ maximizes $P(\mathbf{y}_1)$ so we can write

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1} (P(\mathbf{w}^T \mathbf{x})). \tag{10}$$

Having estimated the first $n-1$ smooth components the next one is calculated as most smooth component of the residual obtained in Gram-Schmidt orthogonalization:

$$\mathbf{w}_n = \arg\max_{\|\mathbf{w}\|=1} (P(\mathbf{w}^T (\mathbf{x} - \sum_{i=1}^{n-1} \mathbf{y}_i \mathbf{y}_i{}^T \mathbf{x}))), \tag{11}$$

where $\mathbf{y}_i = \mathbf{w}_i{}^T \mathbf{x}, i = 1 \dots n$. As the numerical algorithm for finding $\mathbf{w}_n$ we can employ the conjugate gradient method with golden section as a line search routine. The algorithm outline for initial $\mathbf{w}_i(0) = rand$, $\mathbf{p}_i(0) = -\mathbf{g}_i(0)$ is as follows:

1. Identify the indexes l for extreme signal values:

$$l^{\max} = \arg\max_{l \in 1 \dots N} \mathbf{w}_i{}^T(k)\mathbf{x}(l), \tag{12}$$

$$l^{\min} = \arg\min_{l \in 1 \dots N} \mathbf{w}_i{}^T(k)\mathbf{x}(l), \tag{13}$$

2. Calculate gradient of $P(\mathbf{w}_i{}^T\mathbf{x})$:

$$\mathbf{g}_i = \frac{\partial\, P(\mathbf{w}_i{}^T\mathbf{x})}{\partial\, \mathbf{w}_i} = \frac{\sum_{l=2}^{N}\Delta\mathbf{x}(l)\cdot sign(\mathbf{w}_i{}^T\Delta\mathbf{x}(l)) - P(\mathbf{w}_i{}^T\mathbf{x})\cdot(\mathbf{x}(l^{\max}) - \mathbf{x}(l^{\min}))}{\max(\mathbf{w}_i{}^T\mathbf{x}) - \min(\mathbf{w}_i{}^T\mathbf{x}) + \delta(\max(\mathbf{w}_i{}^T\mathbf{x}) - \min(\mathbf{w}_i{}^T\mathbf{x}))}, \tag{14}$$

where $\Delta\mathbf{x}(l) = \mathbf{x}(l) - \mathbf{x}(l-1)$,

3. Identify the search direction (Polak-Ribiere formula[19])

$$\mathbf{p}_i(k) = -\mathbf{g}_i(k) + \frac{\mathbf{g}_i{}^T(k)(\mathbf{g}_i(k) - \mathbf{g}_i(k-1))}{\mathbf{g}_i{}^T(k-1)\mathbf{g}_i(k-1)}\mathbf{p}_i(k-1)\,, \tag{15}$$

4. Calculate the new weights:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \alpha(k)\cdot\mathbf{p}_i(k)\,, \tag{16}$$

where $\alpha(k)$ is found in golden search.

The above optimization algorithm should be applied as a multistart technique with random initialization [14].

4 Component Classification

After latent component are estimated by e.g. SmCA we need to label them as destructive or constructive. The problem with proper signal classification can be difficult task because obtained components might be not pure constructive or destructive due to many reasons like improper linear transformation assumption or other statistic characteristics than explored by chosen BSS method [21]. Consequently, it is possible that some component has constructive impact on one model and destructive on the other. There may also exist components destructive as a single but constructive in a group. Therefore, it is advisable to analyze each subset of the components separately. In particular, we eliminate each subset (use the matrix $\hat{\mathbf{A}}$) and check the impact on the final results. Such process of component classification as destructive or constructive is simple and works well but for many components it is computationally extensive.

5 Generalized Mixing

As was mentioned above, the latent components can be not pure so their impact should have weight other than 0. It means that we can try to find the better mixing system than described by $\hat{\mathbf{A}}$. The new mixing system can be formulated more general than linear, e.g. we can employ MLP neural network:

$$\widehat{\mathbf{X}} = \mathbf{g}^{(2)}(\mathbf{B}^{(2)}[\mathbf{g}^{(1)}(\mathbf{B}^{(1)}\mathbf{S} + \mathbf{b}^{(1)})] + \mathbf{b}^{(2)})\,, \tag{17}$$

where $\mathbf{g}^{(i)}(.)$ is a vector of nonlinearities, $\mathbf{B}^{(i)}(.)$ is a weight matrix and $\mathbf{b}^{(i)}(.)$ is a bias vector respectively for i-th layer, i=1,2. The first weight layer will produce results related to (4) if we take $\mathbf{B}^{(1)} = \hat{\mathbf{A}}$. But we employ also some nonlinearities and the second layer, so in comparison to the linear form the mixing system gains some flexibility. If we learn the whole structure starting from system with initial weights of $\mathbf{B}^{(1)}(0) = \hat{\mathbf{A}}$, we can expect the results will be better, see Fig. 2.

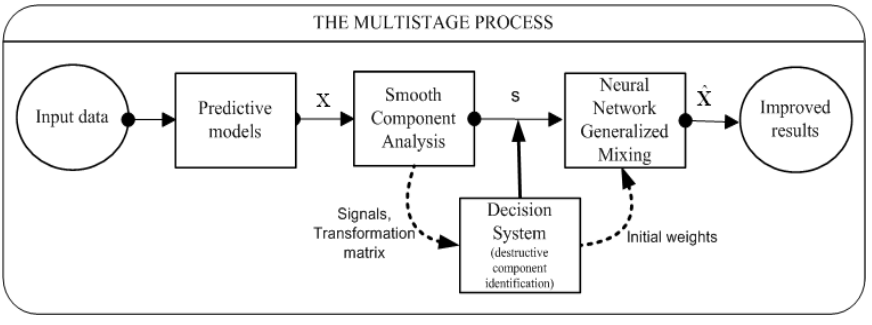

Fig. 2. The concept of filtration stage

6 Electricity Consumption Forecasting

The tests of proposed concept were performed on the problem of energy load prediction [11]. Our task was to forecast the hourly energy consumption in Poland in 24 hours basing on the energy demand from last 24 hours and calendar variables: month, day of the month, day of the week, and holiday indicator. We learned six MLP neural networks using 1851 instances in training, 1313 – in validation, and 1313 – in testing phase. The networks have the structure: M1=MLP(5,12,1) M2=MLP(5,18,1), M3=MLP(5,24,1), M4=MLP(5,27,1), M5=MLP(5,30,1), M6=MLP(5,33,1), where in parenthesis you can find the number of neurons in each layer. The quality of the results is measured with Mean Absolute Percentage Error:

$$MAPE = \frac{1}{N} \cdot \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \tag{18}$$

where i is the index of observation, N- number of instances, y_i - real load value, and $\hat{y}_i$ - predicted value.

In Table 2 we can observe the MAPE values for primary models, effects of improving the modelling results with particular decomposition, and with decomposition supported by neural networks remixing. The last column in Table 2 shows percentage improvement of the best results from each method versus the best primary result.

Table 2. Values of MAPE for primary models and after

Methods	Models						Best result	%
	M1	M2	M3	M4	M5	M6		
Primary results	2.392	2.365	2.374	2.402	2.409	2.361	2.361	-
Decorr.	2.304	2.256	2.283	2.274	2.255	2.234	2.234	5.4
Smooth	2.301	2.252	2.357	2.232	2.328	2.317	2.232	5.5
ICA	2.410	2.248	2.395	2.401	2.423	2.384	2.248	4.8
Decorr&NN	2.264	2.241	2.252	2.247	2.245	2.226	2.226	5.7
Smooth&NN	2.224	2.227	2.223	2.219	2.232	2.231	2.219	6.0
ICA&NN	2.327	2.338	2.377	2.294	2.299	2.237	2.237	5.3

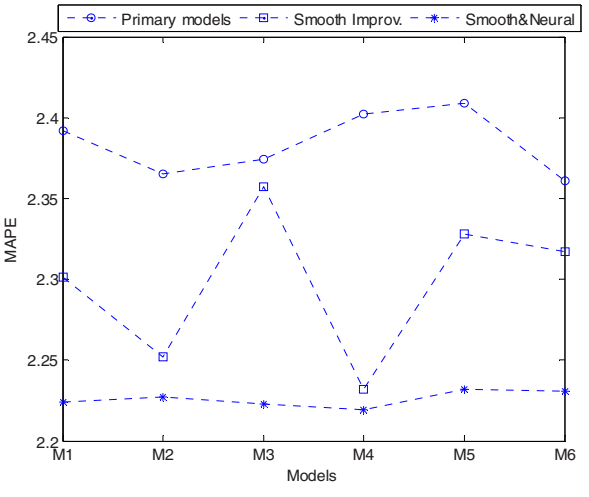

Fig. 3. The MAPE for primary models, improvement with SmCA, and improvement by SmCA&NN

To compare the obtained results with other ensemble methods we applied also bagging and boosting techniques for the presented problem of energy load prediction. They produced predictions with MAPE of 2.349 and 2.226, respectively, what means results slightly worse than SmCA with neural generalisation.

7 Conclusions

The Smooth Component Analysis as well as the other Blind Signal Separation methods can be successfully used as a novel methodology for prediction improvement. The practical experiment with the energy load prediction confirmed the validity of our method. Due to lack of space we compare SmCA approach only with basis BSS methods like decorrelation and ICA. For the same reason extended comparison with other ensemble methods was left as the further work.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
2. Cardoso, J.F.: High-order contrasts for independent component analysis. Neural Computation 11, 157–192 (1999)
3. Choi, S., Cichocki, A.: Blind separation of nonstationary sources in noisy mixtures. Electronics Letters 36(9), 848–849 (2000)
4. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing. John Wiley, Chichester (2002)
5. Donoho, D.L., Elad, M.: Maximal Sparsity Representation via l1 Minimization. The Proc. Nat. Acad. Sci. 100, 2197–2202 (2003)
6. Golub, G.H., Van-Loan, C.F.: Matrix Computations. Johns Hopkins, Baltimore (1996)
7. Haykin, S.: Neural nets: a comprehensive foundation. Macmillan, NY (1994)
8. Hoeting, J., Mdigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging: a tutorial. Statistical Science 14, 382–417 (1999)
9. Hurst, H.E.: Long term storage capacity of reservoirs. Trans. Am. Soc. Civil Engineers 116 (1951)
10. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley, Chichester (2001)
11. Lendasse, A., Cottrell, M., Wertz, V., Verdleysen, M.: Prediction of Electric Load using Kohonen Maps – Application to the Polish Electricity Consumption. In: Proc. Am. Control Conf. Anchorage AK, pp. 3684–3689 (2002)
12. Lee, D.D., Seung, H.S.: Learning of the parts of objects by non-negative matrix factorization. Nature, 401 (1999)
13. Mitchell, T.: Machine Learning. McGraw-Hill, Boston (1997)
14. Scales, L.E.: Introduction to Non-Linear Optimization. Springer, NY (1985)
15. Stone, J.V.: Blind Source Separation Using Temporal Predictability. Neural Computation 13(7), 1559–1574 (2001)
16. Szupiluk, R., Wojewnik, P., Zabkowski, T.: Model Improvement by the Statistical Decomposition. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 1199–1204. Springer, Heidelberg (2004)
17. Therrien, C.W.: Discrete Random Signals and Statistical Signal Processing. Prentice Hall, New Jersey (1992)
18. Yang, Y.: Adaptive regression by mixing. Journal of American Statistical Association, 96 (2001)
19. Zibulevsky, M., Kisilev, P., Zeevi, Y.Y., Pearlmutter, B.A.: BSS via multinode sparse representation. Adv. in Neural Information Proc. Sys. 14, 185–191 (2002)