NAP-SC: A Neural Approach for Prediction over Sparse Cubes

Wiem Abdelbaki[1], Sadok Ben Yahia[1,2], and Riadh Ben Messaoud[3]

[1] Faculty of Sciences of Tunis, El-Manar University, 2092 Tunis, Tunisia
[2] Institut Mines-TELECOM, TELECOM SudParis, UMR CNRS Samovar, 91011 Evry Cedex, France
[3] Faculty of Economics and Management of Nabeul, Carthage University, 1054 Tunis, Tunisia

Abstract. OLAP techniques provide efficient solutions to navigate through data cubes. However, they are not equipped with frameworks that empower user investigation of interesting information. They are restricted to exploration tasks.

Recently, various studies have been trying to extend OLAP to new capabilities by coupling it with data mining algorithms. However, most of these algorithms are not designed to deal with sparsity, which is an unavoidable consequence of the multidimensional structure of OLAP cubes.

In [1], we proposed a novel approach that embeds Multilayer Perceptrons into OLAP environment to extend it to prediction. This approach has largely met its goals with limited sparsity cubes. However, its performances have decreased progressively with the increase of cube sparsity.

In this paper, we propose a substantially modified version of our previous approach called NAP-SC (Neural Approach for Prediction over Sparse Cubes). Its main contribution consists in minimizing sparsity effect on measures prediction process through the application of a cube transformation step, based on a dedicated aggregation technique.

Carried out experiments demonstrate the effectiveness and the robustness of NAP-SC against high sparsity data cubes.

Keywords: Data Warehouse, Data Mining, Principal Component Analysis, Machine Learning, Multilayer Perceptron, Prediction.

1 Introduction

A Data Warehouse (DW) is a corner stone in the Business Intelligence (BI) process. It is implemented to store analysis contexts within multidimensional data structures, referred to as *Data Cubes* and usually manipulated by using On-line Analytical Processing (OLAP) applications to enable senior managers exploring information and getting BI reportings through interactive dashboards.

By definition, a DW should fundamentally contain integrated data [2]. However, generally, exploring a data cube disclose a sparse structure within several empty measures. In DW models, empty measures correspond to non-existent facts reflecting either out-of-date events that did not happen or upcoming events

S. Zhou, S. Zhang, and G. Karypis (Eds.): ADMA 2012, LNAI 7713, pp. 340–352, 2012.

that have not yet occurred and may happen in the future. We argue that predicting these measures could consolidate BI reportings and provide new opportunities to DW customers by enlarging their dashboard picture. For instance, it will be extremely useful to a retailer chain to predict the potential sale amount of ice cream in January in some particular agencies. This indicator will definitely help the company to optimise the number of ice cream freezers to install in that period.

So far, non-existent measures in a data cube may potentially be learned from its neighborhood. Agarwal and Chen state that making future decisions over historical data is one crucial goal of OLAP [3]. However, OLAP is restricted to exploration and not equipped with a framework to empower user investigation of interesting information. In fact, despite the fundamental Cood's statement of goal seeking analysis models (such as "What if" analysis) required in OLAP applications since the early 90's [4], most of today's OLAP products still lack an effective implementation of this feature.

On the other hand, data mining is a mature, robust field that have proven its efficiency in dealing with complex data sets [5]. Recently, several approches have been attempting to perform data mining techniques on OLAP cubes. They tackled several issues like cube exploration [6], association rules mining [7] and prediction [8,3]. In [1], we adopted this approach while attempting to predict non-existent measures over OLAP cubes. Thus, we attempted to adapt neural networks, which are among the most popular machine learning techniques that have been explored to solve data mining problems [5], to OLAP environment. To that end, we proposed a neural based approach to predict measures over high-dimensional data cubes that we consider as a Neural Approach to Prediction over High-dimensional Cubes (NAP-HC).

The experimental study showed that NAP-HC has largely met its goals in the case of limited sparsity data cubes. However, its performances decrease within sparse data cubes. This deterioration is justifiable, since various researches affirm that sparsity affects the performances of any approach trying to combine OLAP with data mining methods [9,10]. Moreover, sparsity is an unavoidable consequence of the multidimensional structure of data cubes. It is generated by relationships between dimension attributes. For example, while investigating some product sales in a retail chain according to time and store location, drilling down the location dimension to departments level will disclose many empty cells, generated by the departments that do not sell that product from the first. Kang *et al.* argue that this case appears very often due to OLAP applications' nature of business [11].

In this paper, we introduce a novel Neural Approach for Prediction over Sparse Cubes (NAP-SC). We stress that our current proposal does not upgrade NAP-HC, which we still recommend for limited sparsity cubes. However, it is an alternative version designed for the particular case of high sparsity data cubes, which makes the following contributions:

- Getting more value out of our recently proposed approach [1], by further exploring its framework and techniques.
- Minimizing sparsity effect on analysis by embedding in a cube transformation step, based on a dedicated aggregation technique.
- Involving the hierarchical structure of data cubes in the analysis to enable the prediction system to deal with multiple hierarchical levels at once.

This paper is organized as follows. In Section 2, we expose a state of the art of works related to predictions in data cubes. In Section 3, we present a reminder of NAP-HC and define our analysis context. Section 4 details the method formalization that we followed in our proposal. In Section 5, we carry out experiments investigating the effectiveness of NAP-SC. Finally, Section 6 summarizes our findings and addresses future research directions.

2 Related Work

Performing data mining tasks on DWs represents an important topic in DW technology. Goil and Choudhary argue that data mining automated techniques further empower OLAP and make it more useful [12]. Several researches were proposed under different motivations (discovery-driven exploration of cubes, cube mining, cube compression, and so on). In line with our concern, we focus on those having a close linkage with prediction in DWs.

We summarize in Table 1 proposals attempting to extend OLAP to prediction. They are detailed according to seven main criteria: (1) What is the overall goal of the proposal? (2) Does it include an algorithmic optimization? (3) Does it use a reduction technique? (4) Does it introduce new classes of measures? (5) Does it provide explicit predicted values of non-existent measures? (6) Does the proposal involve the hierarchical structure of data cubes in the analysis? and(7) Does it deal explicitly with sparsity? We note (+) if the proposal meets the criteria, and (-) if not.

Palpanas *et al.* used the principle of information entropy to build a probabilistic model capable of detecting measure deviations to compress data cubes [13].

Table 1. Proposals integrating prediction in data cubes

Proposal	Goal	Optimization	Reduction	Measures	Values	Hierarchies	Sparsity
[13]	Compression	-	+	-	-	+	-
[14]	Compression	-	+	+	-	+	-
[6]	Exploration	+	-	-	-	+	-
[15]	Prediction	+	-	+	-	+	-
[8]	Prediction	+	-	-	-	-	-
[3]	Prediction	+	-	+	-	+	-
[1]	Prediction	-	+	+	+	-	-
NAP-SC	Prediction	+	+	+	+	+	+

Their approach predicts low-level measures from high-level pre-calculated aggregates. Chen *et al.* introduced a new class of data cubes, called *Regression Cubes* [14]. They contain compressible measures indicating the general tendency and variations compared to original ones. Sarawagi *et al.* proposed a log linear model to assist DW users when exploring data cubes by detecting exceptions [6]. Chen *et al.* introduced the concept of *Prediction Cubes*, where a score or a distribution of probabilities of measures are fetched beside their original values [15]. They are used to build prediction models. Bodin-Niemczuk *et al.* proposed to equip OLAP with a regression tree to predict measures of forthcoming facts [8]. Agarwal and Chen introduced a new data cube class called *Latent-Variable Cube* [3]. It is able to compute aggregate functions, such as mean and variance, over latent variables detected by a statistical model.

In [1], we proposed NAP-HC that predicts measures over high-dimensional data cubes. It introduces a new class of cubes, called PCA-cubes, integrating customized measures referring to predictors stored in an external database. The approach operates on two main stages. The first is a pre-processing one that makes use of the Principal Component Analysis (PCA) to reduce data cube dimensionality. As for the second stage, it introduces an OLAP oriented architecture of Multilayer Perceptrons (MLP)s that learns from multiple training-sets without merging them, and yet comes out with unique predicted value for each targeted measure.

From the above cited references, an outstanding common observation is dealing with data dimensionality. Indeed, the multidimensional structure of data and the usual huge facts' volumetry in DWs represent one of the most challenging issues of integrating predictive models into OLAP environment. This could be of a negative effect on prediction performance, which is supposed to provide BI reporting costumers with fast and accurate results in line with OLAP applications. Thus, some of the above proposals rely on heuristics to optimize implemented algorithms [6,15,3]. Some others rather consider a pre-processing stage to reduce dimensionality effect on their algorithms [13,14,1].

One of the most fundamental challenges of associating OLAP with a predictive model concerns involving the hierarchical structure of data cubes to improve analysis performances. While some approaches employ low level model to constitute higher aggregation level models as [6], other approaches inverse this methodology to derive, low-level facts from their existent aggregates as [13]. NAP-HC does not handle multiple hierarchical levels. It explores one level per dimension during all the analysis. Nevertheless, NAP-SC explores multiple hierarchical levels during the dimension reduction and the prediction stages.

On the other hand, sparsity, which is the case of most of OLAP cubes, remains a very serious issue. Thomsen defines it as the degrees to which cells contain invalid values instead of data [16]. In addition of increasing the access time, it degrades most of analysis techniques applied on data cubes [9,10]. For instance, the conducted experiments in [1] showed clearly that the performances of NAP-HC decreased progressively while increasing the sparsity level of the treated cube. Therefore, we argue that handling sparsity represents a fundamental challenge

for any approach trying to extend data mining algorithms to OLAP environment. Despite this, all of the above cited researches do not provide explicit solutions of their confrontations with sparsity.

We affirm that the solutions reached in [1] still hold a lot of promises and major aspects of their potential contributions remain unexplored. Therefore, we propose to study them further by proposing NAP-SC, which is a novel approach for predicting non-existent measures over sparse OLAP cubes. It has the general overview of NAP-HC since the two approaches operate according to the same global stages. Nevertheless, a closer look reveals substantial differences that we will expose in the following sections of this paper.

3 General Notations

In this section we present the general notations that we use in this paper. We also present the definitions introduced in [1] accompanied with a short reminder of NAP-HC. We entend to reuse the same data cube definition provided in [7].

Let $\mathcal{C}$ be a data cube having the following proprieties:

- $\mathcal{C}$ has a non empty set of d dimensions $\mathcal{D} = \{D_i\}_{(1 \leq i \leq d)}$;
- $\mathcal{C}$ contains a non empty set of m measures $\mathcal{M} = \{M_q\}_{(1 \leq q \leq m)}$;
- Each dimension D_i contains l_i categorical attributes;
- H_i is the set of hierarchical levels of the dimension D_i. H_j^i is the j^{th} hierarchical level in D_i.

We also use the concept of *cube level*, proposed by Agarwal and Chen in [3]. It defines a vector of distinct dimensions levels.

NAP-HC embeds MLPs, which had proven their performances in prediction tasks [17,18], within OLAP environment in a two stage proposal. The first stage consists in generating reduced information preserving training sets over the original cube while preserving the measure variations and the semantics linking attributes and dimensions. In order to do so, we resorted Principal Component Analysis (PCA) as a dimensions reduction technique [19]. We exploited its orthogonal transformation to convert the correlated dimension attributes into smaller sets of principal components.

As PCA is not designed for multidimensional structures, we introduced the concept of cube-face to identify all the possible configurations of the data cube and cover all the measure's variations.

Definition 1 (Cube-face). *Let* $\{D_k, D_v, D_{s_1}, \ldots, D_{s_f}, \ldots, D_{s_{d-2}}\}_{(1 \leq f \leq d-2)}$ *be a non-empty subset of* d *distinct dimensions.*

We denote by $Cf(D_k, D_v, (D_{s_1}, \ldots, D_{s_f}, \ldots, D_{s_{d-2}}))$ *a cube-face* Cf *of a data cube* $\mathcal{C}$*. It is a data view of a data cube identifiable by the geometrical positions of its dimensions that we call: Key dimension* D_k*, Variant dimension* D_v *and a set of* $(d-2)$ *Slicer dimensions* $D_{s_{d-2}}$*.*

The number, n, of extractable cube-faces over a data cube is equal to the number of its geometrical faces.

To preserve the semantics linking attributes and dimensions, we introduced the concept of PCA-slice.

Definition 2 (PCA-slice). $P(D_k, D_v, (a^o_{t_1}, \ldots, a^p_{t_f}, \ldots, a^q_{t_{d-2}}))$ *is the PCA-slice obtained by applying the OLAP* Slice *operator on* Cf*; with* $a^o_{t_1} a^p_{t_f}$ *and* $a^q_{t_{d-2}} \in D_{s_1}, D_{s_f}$ *and* $D_{s_{d-2}}$*, respectively.*

The coordinate factors are generated by iteratively applying PCA on the extracted PCA-slices and stored in external tables that we called PCA-tables. In order to track the membership of a measure and its corresponding coordinate factors, we introduced the concept of PCA-cube.

Definition 3 (PCA-cube). *A PCA-cube is a data cube that contains, beside its original measures, a new type of measures consisting of references to the sets of coordinate factors associated to each cell.*

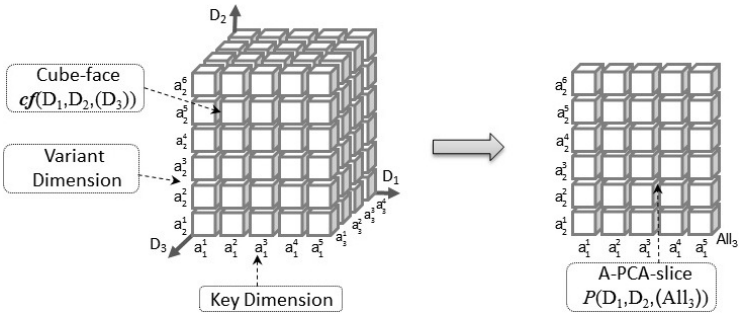

Fig. 1. Cube-face transformation

As for the second stage, we designed a new MLPs architecture to overtake the multidimensional structure of data cubes. It consists of an interconnection of $n+1$ sub-networks. Firstly, n child-networks, each one associated to a distinct cube-face. It gets that cube-face coordinate factors as inputs and provides the targeted measures as output. Then, a combinator-network that receives the outputs of all child-networks as inputs and comes out with the targeted measures. The innovative aspect of this architecture consists in involving multiple training-sets in the same learning process without having to merge them and yet generating a unique predicted value for each targeted measure.

Like NAP-HC, NAP-SC is not a cube completion technique. It is not supposed to fill all empty measures of a data cube. Its main objective is to promptly come out with prediction of any empty measure upon the request of the user.

4 Formalization of Our Proposal

Most of machine learning algorithms are not designed to deal with missing values. Many researches apply deletion techniques, which provide trivial solutions

that enable data mining algorithm application. However, they may seriously affect data quality and lead to non representative data set. Other researches use imputation methods like multiple imputation [20], regression, mean imputation, etc. However, these methods are designed for bi-dimensional data and are not adapted to multidimensional structures.

In [21], Ben Messaoud *et al.* proposed a cube reorganization approach that generates more dense cubes from spare ones. While facing the sparsity issue, the authors proposed to explore OLAP aggregation operators to minimize sparsity effect. They transformed a data cube into a complete disjunctive table by fixing two dimensions, treated as instance and variable dimensions, and aggregating the remaining ones to the highest level, which is naturally the *All* level.

We affirm that aggregating some dimensions to higher levels enable the analysis to avoid many empty cells. However, we notice that fixing one specific combination of dimensions may promote some dimensions on the expense of others. This may cause a loss of the information extractable over the unexplored combinations. That is to say, any transformation of the dimensions combination will certainly lead to a whole different data set that remain unexplored in the case of [21]'s approach.

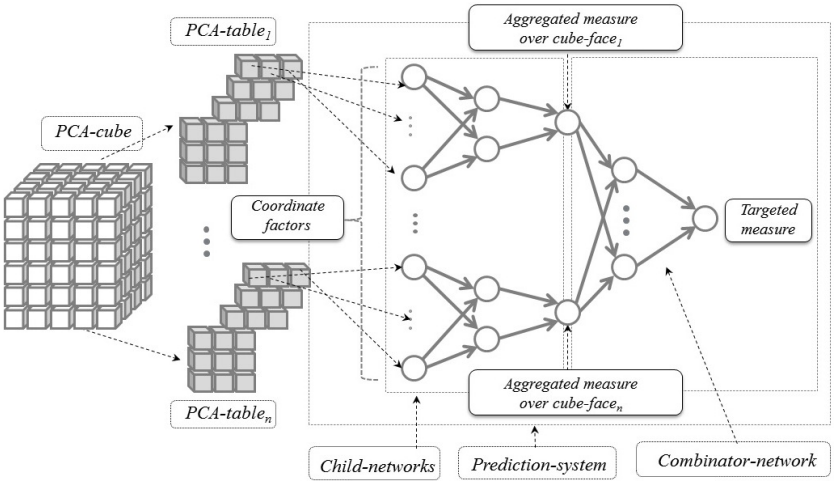

Fig. 2. Prediction system

We intend to revise this transformation technique by coupling it with NAP-HC cube decomposition technique. This will enable it to consider all the extractable dimension combinations over the data cube and thus to treat equitably all dimensions. Then, we will explore the revised version to minimize sparsity effect on cube-faces.

On the other hand, we must consider that the aggregation step causes loss of precision in term of dimension attributes. Nevertheless, we intend to make up for this by revising the forthcoming prediction system and enabling it to involve multiple hierarchical levels in the same prediction process.

4.1 Dimensions Reduction

The main purpose of the dimensions reduction stage is to generate information preserving, concentrated, decorrelated training sets over the original data cube. Like NAP-HC, NAP-SC uses PCA as a dimensions reduction technique and explore its orthogonal transformations to extract smaller sets of principal components. They serves as predictors of the forthcoming prediction system. The sheer novelty of NAP-SC dimensions reduction stage consists in including a revisited version of Ben Messaoud *et al.*'s proposal to minimize sparsity effect.

Actually, the reduction process starts by identifying all the cube-faces from the data cube following the initially selected cube level. At this point, NAP-HC extracts all the PCA-slices from the cube-faces and iteratively applies PCA on them. PCA-slices are generally sparse, what affects the reduction and thus the prediction process. To minimize sparsity effect, we propose to reuse the transformation proposal of Ben Messaoud *et al.* after coupling it with our cube decomposition proposal to preponderate equitability among all the cube dimensions.

Algorithm 1. Training algorithm

Input: $\mathcal{P}cc, cube\_faces'set, RMSE\_max$
Output: $child, combinator, RMSE$

```
1  foreach cube_face in cube_faces'set do
2      initialize(child);
3      converged ← false;
4      while converged = false do
5          agg_m ← extract_agg(cube_face.A_PCA_slice);
6          pc[] ← extract_comp(Pcc, agg_m);
7          propagate(pc, agg_m, child);
8          adjust(child);
9          if RMSE(child) < RMSE_max then
10             converged ← true;
11
12 initialize(combinator);
13 converged ← false;
14 while converged = false do
15     foreach child do
16         agg_m ← extract_agg(cube_face.A_PCA_slice);
17         pc[] ← extract_comp(Pcc, agg_m, cube_face);
18         combinator_input[] ← propagate(pc, agg_m, child);
19     m ← extract_measure(Pcc);
20     propagate(combinator_input[], m, combinator);
21     adjust(combinator);
22     if RMSE(combinator) < RMSE_max then
23         converged ← true;
24         RMSE ← RMSE(combinator);
25         return child networks' set, combinator, RMSE;
```

Thus, we transform each cube-face as follows: We keep the *key* and the *variable* dimensions still. Then, using the aggregation function that had been used in the initial cube computation, we totally aggregate all the *slicer* dimensions to the *All* level, as mentioned in Figure 1. This operation generates one single large PCA-slice per cube-face. To distinguish it from the classical PCA-slice, we call it *Aggregated-PCA-slice* (A-PCA-slice). The generated A-PCA-slices are much less sparse than the classical PCA-slices due to the aggregation step that enable the analysis to avoid many empty cells by aggregating them according to wisely selected dimensions.

Finally, we apply PCA on each A-PCA-slice. This operation generates sets of coordinate factors, which we store in PCA-tables. Then, we reuse the concept of PCA-cube to track the membership of measures and their coordinate factors.

Like most conventional OLAP pre-processing phases, the reduction stage is a time consuming process. Therefore, we believe that it should be executed in back-stage, on a regular basis, by the end of each periodic data loading of the DW.

4.2 Measure Prediction

The main goal of this stage is to apply MLPs, which can not handle multidimensional structures, on OLAP cubes. NAP-HC prediction system consists of an OLAP oriented MLPs architecture that consider multiple disjoint training-sets without having to merge them. And yet, it comes out with a unique predicted value of each targeted measure. We intend to reuse the general aspects of this system with NAP-SC. Nevertheless, we entend to empower it further, in order to make up for the loss of precision caused by the aggregation step. Thus, we preserve the general overview of the MLPs architecture and modify its training algorithm to enable the prediction system to handle multiple cube levels to restore the initial cube level targeted by the user .

As shown in Figure 2, NAP-SC prediction system consists of an interconnection of multiple sub-networks; $n-1$ child-networks and one single combinator-network. Each child-network is associated to one distinct A-PCA-slice, and thus to one cube-face. In addition, for each cube-face, we intentionally emphasized the same dimensions in both of measures concentration and cube-face transformation steps. In such a way, the reduction stage preserves the relationships of original and aggregated measures. Thus we affirm that an appropriate prediction system can come out with the original values from the aggregated ones. In our case, it is NAP-SC's prediction system, whose child-networks consider aggregated measures as target instead of initially selected levels measures. As for its combinator-network, it targets the initially selected levels measures. These transitions between correlated dimensions' levels enable the prediction system to deal with different cube levels and to restore the lower cube-level values targeted by the user from the higher cube levels exploited in the reduction stage.

The training algorithm of the prediction system is provided in Algorithm 1. It uses Root Mean Squared Error (RMSE) as stopping criteria. It requires the PCA-cube, the set of cube-faces and a maximum tolerable value of RMSE as inputs.

Each child-network is initialized by randomizing its internal weights and setting-up its structure according to its associated cube-face dimensions. Then, it is trained using a randomly selected set of cells from the PCA-cube $\mathcal{P}cc$. It gets the coordinate factors referenced by the PCA-cube as inputs and the aggregated measures extracted from the cube-face's A-PCA-slice as output. The training process is repeated iteratively, until convergence.

After all the child-networks are trained, the combinator-network is to be initialized. It gets a number of input nodes equal to the number of child-networks and one single output node. It is trained in a batch mode with the trained child-networks' outputs. Unlike child-networks that consider aggregated measures as outputs, the combinator-network consider the measure derived from the initially targeted cube-level as output. It provides the measures's values targeted by the user.

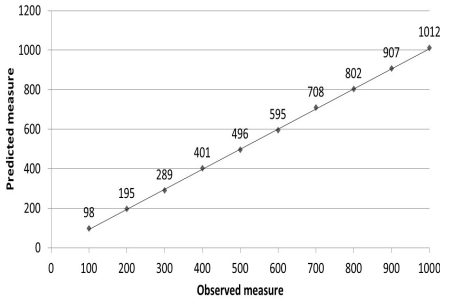

Fig. 3. Prediction quality

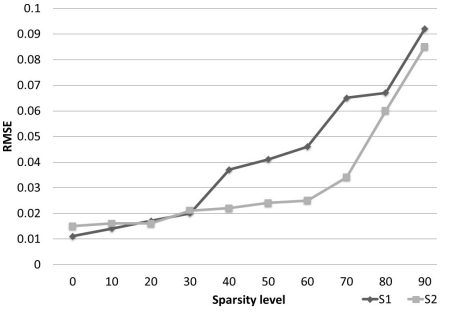

Fig. 4. Performances against sparsity

As several theoretical and empirical studies show that a single hidden layer is sufficient to achieve a satisfactory approximation of any nonlinear function [17], we restrict the architecture of all sub-networks to three layers, including a hidden one. We also use the gradient back-propagation algorithm [22] that has proved its usefulness in several applications [18,17]. We associate it with the conjugate gradient learning method and the sigmoid activation function.

5 Experimental Study

To test our approach, we implemented an experimental prototype of our system. We adapted the *American Community Surveys 2000-2003* [1] database to DW context and exploited it in our experimental study. It is a real-life, derived from the U.S.A census database. It contains samples of the American population treated between 2000 and 2003.

[1] American Community Surveys is accessible from the official site IPUMS-USA (Integrated Public Use Microdata Series); `http://sda.berkeley.edu`

5.1 Analysis Context

We consider a 4 dimensions data cube; *Location*, *Origin*, *Education* and *Time*, with 3.8 million facts. We aim at predicting the number of people of a certain race, according to their cities and their levels of education. To compute the initial cube level, we focus on the *person count* measure and select the hierarchical levels; *State*, *Race* and *Education*. These levels contain, respectively, 51, 10 and 14 dimension attributes. After the application of our reduction approach, we end up with 6 cube-faces that generate 10, 12, 4, 3, 4 and 3 principal components. For the prediction phase, we use the 10-fold cross-validation technique and the RMSE as a quality indicator. Our experimental study is spread over two experiments to investigate the following aspects.

5.2 Prediction Quality

The purpose of the first experiment is to investigate the performances of NAP-SC in the case of real-life high sparsity data cube. We elaborated a predictive system that faithfully represents our proposed architecture. We have set the hidden neurons number of each sub-network hidden layer, to one half of the number of its own input. Then, we increased the sparsity of our treated cube by deleting 30% of its fact table records.

After training of our system, we tried to predict a set of random measures that had not been included in the learning process. To properly present the results, we considered a separated by regular intervals set of measure values. We presented the resulting curve in Figure 3. We note that the predicted values have minimum distances from the line *observed measure = predicted measure*. Furthermore, the correlation coefficient of these values is equal to 0.96, what indicates an accurate prediction.

5.3 Efficiency against Sparsity

In order to investigate NAP-SC efficiency against sparsity, we compared its performances with those of NAP-HC, while varying the sparsity level. Thus, we started with our initial data cube and increased the sparsity of the data cube by 10% at each time.

We present the results in Figure 4, $S1$ and $S2$ translate, respectively, the performances of NAP-HC and NAP-SC. We notice that NAP-HC outperformed NAP-SC for a level of sparsity between 0% and 25%. From 30%, NAP-SC takes the lead and outperforms NAP-HC. Then, it preserves its robustness until a percentage of 70% of non-existent facts. However, from 70%, NAP-SC's RMSE evolution becomes important and the prediction quality decreases remarkably. This is explained by the fact that the minimum number of valid instances becomes insufficient from a level of sparsity of 70% .

Through this experiment, we highlighted the usefulness of both NAP-HC and NAP-SC. The BI customer can use either system depending on his treated cube.

We affirm that he can even apply individually both systems on the same data cube according to the sparsity level of the manipulated dimensions' hierarchical levels.

6 Conclusion and Perspectives

In this paper, we explored neural networks, which have been proven their efficiency in several data mining techniques, to empower OLAP and extend it to prediction capabilities.

We proposed NAP-SC that predicts non-existent measures over sparse data cubes. It is a substantially modified version of our previously proposed approach NAP-HC [1], designed to solve the same problem in the case of high dimensional non sparse data cubes. The main contribution of NAP-SC consists in minimizing sparsity effect on analysis. It is ensured through a cube transformation step based on a dedicated aggregation technique. In addition, the prediction stage of NAP-SC involves multiple cube levels in the same prediction process, what embeds further our proposed MLPs architecture into OLAP environment.

The experimental study showed the improved accuracy of NAP-SC and its robustness against sparsity. Notwithstanding, NAP-HC outperformed NAP-SC in the case of limited sparsity data cubes. Thus we conclude that both system are useful for different scenarios. It is up to BI customer to choose between them according to the nature of his treated cubes. He can even apply individually both systems on the same data cube in different cube levels.

As part of future work, we plan to formalize explicit criterion indicating which system to apply between NAP-HC and NAP-SC. In addition, we plan to include a framework that explains the reasons of non-existent measures occurrences, similarly to that of [23]. Finally, we aim at modeling a theoretical relation between the reduction and the prediction stage to optimize our model.

References

1. Abdelbaki, W., Ben Messaoud, R., Ben Yahia, S.: A Neural-Based Approach for Extending OLAP to Prediction. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 117–129. Springer, Heidelberg (2012)
2. Inmon, W.H.: Building the Data Warehouse. John Wiley & Sons (1996)
3. Agarwal, D., Chen, B.C.: Latent OLAP: Data Cubes Over Latent Variables. In: Proceedings of the 2011 International Conference on Management of Data, SIGMOD 2011, pp. 877–888. ACM, New York (2011)
4. Codd, E.F., Codd, S.B., Salley, C.T.: Providing OLAP (on-line Analytical Processing) to User-analysts: An IT Mandate, vol. 32. Codd & Date, Inc. (1993)
5. Olson, D., Delen, D.: Advanced Data Mining Techniques. Springer (2008)
6. Sarawagi, S., Agrawal, R., Megiddo, N.: Discovery-Driven Exploration of OLAP Data Cubes. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 168–182. Springer, Heidelberg (1998)
7. Ben Messaoud, R., Loudcher-Rabaseda, S.: OLEMAR: An On-Line Environment for Mining Association Rules in Multidimensional Data. In: Advances in Data Warehousing and Mining, vol. 2. Idea Group Publishing (2007)

8. Bodin-Niemczuk, A., Ben Messaoud, R., Rabaséda, S.L., Boussaid, O.: Vers l'intégration de la prédiction dans les cubes OLAP. In: EGC, pp. 203–204 (2008)
9. Niemi, T., Nummenmaa, J., Thanisch, P.: Normalising OLAP Cubes for Controlling Sparsity. Data & Knowledge Engineering 46(3), 317–343 (2003)
10. Kriegel, H.P., Borgwardt, K.M., Kröger, P., Pryakhin, A., Schubert, M., Zimek, A.: Future Trends in Data Mining. Data Min. Knowl. Discov. 15(1), 87–97 (2007)
11. Juyoung, K., Hwanseung, Y., Yoshifumi, M.: Classification of Sparsity Patterns and Performance Evaluation in OLAP Systems. IPSJ SIG Notes 67, 449–455 (2002)
12. Goil, S., Choudhary, A.: High Performance Multidimensional Analysis and Data Mining. In: Proceedings of the High Performance Networking and Computing Conference (SC 1998), Orlando, Florida, USA (November 1998)
13. Palpanas, T., Koudas, N., Mendelzon, A.: Using Datacube Aggregates for Approximate Querying and Deviation Detection. IEEE Trans. on Knowl. and Data Eng. 17, 1465–1477 (2005)
14. Chen, Y., Dong, G., Han, J., Pei, J., Wah, B.W., Wang, J.: Regression Cubes with Lossless Compression and Aggregation. IEEE Trans. on Knowl. and Data Eng. 18 (December 2006)
15. Chen, B.C., Chen, L., Lin, Y., Ramakrishnan, R.: Prediction Cubes. In: Proceedings of the 31st International Conference on Very large Data Bases, VLDB 2005, pp. 982–993 (2005)
16. Thomsen, E.: Olap Solutions: Building Multidimensional Information Systems. Wiley Computer Publishing (2002)
17. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks are Universal Approximators. Neural Networks 2(5), 359–366 (1989)
18. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice Hall International Editions Series. Prentice Hall (1999)
19. Hotelling, H.: Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology 24(7), 498–520 (1933)
20. Rubin, D.B.: Multiple Imputations in Sample Surveys: a Phenomenological Bayesian Approach to Nonresponse. In: Proceedings of the Survey Research Methods Section, pp. 20–28 (1978)
21. Ben Messaoud, R., Boussaid, O., Rabaséda, S.L.: Using a Factorial Approach for Efficient Representation of Relevant OLAP Facts. In: Proceedings of the 7th International Baltic Conference on Databases and Information Systems (DB&IS 2006), pp. 98–105. IEEE Communications Society, Vilnius (2006)
22. Rumelhart, D., McClelland, J., University of California, S.D.P.R.G: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Foundations. Computational Models of Cognition and Perception. MIT Press (1986)
23. Ben Othman, L., Rioult, F., Ben Yahia, S., Crémilleux, B.: Missing Values: Proposition of a Typology and Characterization with an Association Rule-Based Model. In: Pedersen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK 2009. LNCS, vol. 5691, pp. 441–452. Springer, Heidelberg (2009)