PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum
Published by

Publisher:	Rector of the Ilmenau University of Technology Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor:	Marketing Department (Phone: +49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de) Faculty of Computer Science and Automation (Phone: +49 3677 69-2860) Univ.-Prof. Dr.-Ing. habil. Jens Haueisen
Editorial Deadline:	20. August 2010
Implementation:	Ilmenau University of Technology Felix Böckelmann Philipp Schmidt

USB-Flash-Version.

Publishing House:	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16 98693 Ilmenau
Production:	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts
Order trough:	Marketing Department (+49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher:	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

PREDICTING PATTERN-BASED TIME SERIES USING MODELS DERIVED FROM STATISTICAL DATA COMPRESSION

C. Mattern, P. Bretschneider

Fraunhofer IOSB-AST, Am Vogelherd 50, 98693 Ilmenau,
Phone: +49 3677 461-124/-102, E-Mail: peter.bretschneider@iosb-ast.fraunhofer.de

ABSTRACT

This paper investigates the possibility of transferring forecasting methods motivated by data compression into the domain of pattern based time series prediction. The investigation is conducted on the example of short term load forecast. Pattern-based time series are characterized by typical, repeating signal subsequences. Statistical data compression relies on pattern matching and the forecast of future patterns' probabilities. Both, statistical compression and time series prediction, require the estimation of real valued entities. Context Mixing (CM), a subclass of statistical compression algorithms, combines multiple predictions and achieves outstanding compression performance. To transfer CM concepts to time series prediction, the CM model structure is examined. A sample of electrical load data is analyzed to identify its typical characteristics. Subsequently a CM forecasting system is derived, taking the data properties into account. A simulation and an evaluation of the prediction error indicate very accurate forecasts.

Index Terms – short term load forecast, time series forecast, pattern based time series, context mixing

1. INTRODUCTION

The forecast of time series has a widespread application in many different fields of science including economics, ecology and engineering. An estimation of future events outcome often serves as a foundation for planning, decision support and optimization. For instance, modern load management requires the forecast of electrical demand and the fluctuating solar and wind energy output. Over the last decade a rich set of generic models, mostly based on Neural Networks (NN), Fuzzy Systems (FS), Neuro-Fuzzy Systems (NFS) and Support Vector Regression (SVR) have been applied to real-valued prediction problems (e.g. short term load forecasting) [1].

In the field of data compression, statistical algorithms have always shown best compression performance among others. Similar to time series forecast, statistical compression requires models to estimate real values – probabilities. Starting in 2002, series of PAQ data compression programs, developed by Matt Mahoney and modified by many others, evolved and introduced CM as a new statistical compression algorithm [2], [3]. Despite its high processing and memory requirements CM gained growing attention due to its superb performance [4].

Breaking time series forecast and statistical compression down into their simplest element – the estimation of real values – clearly reveals a similarity between them. Starting from that observation this work shows how the concepts belonging to CM can be ported to the domain of time series forecast. Load forecasting, as an important practical example, serves as a test case for CM modeling, simulation and evaluation. The electrical load, typically containing repeating signal subsequences, is pattern based.

This paper is divided into five further sections. The next section introduces the general structure and principles of CM starting from the data compression perspective. In Section 3 an electrical load data sample is analyzed to obtain its typical characteristics, which are utilized to derive a CM based model in section 4. Section 5 shows a simulation and evaluation of the CM model. Finally section 6 summarizes and interprets the work's results.

2. CONTEXT MIXING

2.1. Data compression

Lossless data compression can be formally described as transforming a symbol sequence $s_1 s_2 s_3 \dots s_k$ over a finite alphabet Σ, $s_i \in \Sigma$, into a less redundant representation. Statistical compression processes a single symbol s_k at a time and divides the coding process into two steps: modeling and encoding [5].

In the k^{th} step the (data) model assigns probabilities $P(s_k = s)$ to each possible value $s \in \Sigma$, which are afterwards mapped to a corresponding encoding, typically via arithmetic coding. Probable symbols receive shorter encodings to achieve compression. The prediction accuracy can be improved by taking a discrete context, usually formed by preceding symbols, into account, e.g. an order-N context $P(s_k = s | s_{k-1} s_{k-2} \dots s_{k-N})$. That means a simple predictor $P(s)\ \forall s \in \Sigma$ is addressed based on a discrete context, which is termed context modeling.

Data processing is done bitwise in CM [3], $\Sigma = \{0, 1\}$. Hence estimating $p_k = P(s_k = 1)$ is sufficient, since $P(s_k = 0) = 1 - P(s_k = 1)$. A CM

model mixes M individual predictions p_k^i $1 \leq i \leq M$ using a mixing function m.

$$p_k = m(p_k^1, p_k^2, \dots, p_k^M) \tag{1}$$

2.2. **General structure**

As previously stated CM is a composite approach, i.e. the forecast system consists of at least several individual prediction submodels and a mixing function to combine the submodels' predictions. Each submodel component usually [3]

- works independent of other components,
- is specialized to distinct situations (via contexts) and
- adapts to the processed data.

When mixing the predictions it is important to distinguish how a subset of models is specialized to the current situation, since specialized models are likely to give better predictions. Typically, situations are identified by a discrete context, thus it can be beneficial to select a mixing function based on a context, too.

Substituting the probability estimations p_k^i from eqn. (1) with expected value estimations x_k^i yields the basic structure of a CM model for time series prediction.

$$x_k = m(x_k^1, x_k^2, \dots, x_k^M) \tag{2}$$

Now the questions of choosing a mixing function m and designing the expected value estimations need to be addressed.

2.3. **Basic modeling**

In compression, the minimization of the encoded messages length, i.e. its entropy, is meaningful. The aim of optimal prognosis can be formulated as the minimization of the MSE.

$$\min \frac{1}{L}\sum_{l=1}^{L}(x_l - \hat{x}_L)^2 \tag{3}$$

Eqn. (3) evaluates the mean squared error of a single prediction $\hat{x}_L$ regarding the encountered sequence of L observations $x_1 x_2 \dots x_L$, where x_L is the most recent observation.

Giving higher weights to more recent observations can improve the estimation of $\hat{x}_L$. Thus an exponentially decaying weight

$$c_l = c^{L-l} \tag{4}$$

with $c \in [0, 1) \subset \mathbb{R}$ is used to modify eqn. (3).

$$\min \frac{1}{L}\sum_{l=1}^{L} c_l(x_l - \hat{x}_L)^2 \tag{5}$$

Deriving eqn. (5) and finding its roots with respect to $\hat{x}_L$ results in the optimal estimation.

$$\hat{x}_L = \frac{S_L}{T_L} = \frac{\sum_{l=1}^{L} c_l x_l}{\sum_{l=1}^{L} c_l} \tag{6}$$

A further consideration regarding S_L and eqn. (4) shows, that the estimation can be formulated recursively.

$$S_L = x_L + c\,\underbrace{(x_{L-1} + c\,x_{L-2} + \dots + c^{L-2}x_1)}_{S_{L-1}} \tag{7}$$

The term T_L can be treated in the same fashion. Thus eqn. (6) can be expressed recursively, which eases an implementation.

$$\hat{x}_L = \frac{S_L}{T_L} = \frac{cS_{L-1} + x_{L-1}}{cT_{L-1} + 1} \tag{8}$$

An interpretation of such an estimate can be derived when comparing $\hat{x}_L$ and $\hat{x}_{L-1}$ to yield an adjustment.

$$\hat{x}_L - \hat{x}_{L-1} = \frac{1}{T_L}\underbrace{(x_L - \hat{x}_{L-1})}_{e_L} \tag{9}$$

Eqn. (9) depicts an adjustment proportional to the prediction error e_L. As L increases, the term $1/T_L$ decreases. For a very large number of observations, i.e. $L \to \infty$, the geometric series T_L converges to its asymptotic limit.

$$\frac{1}{T_L} \xrightarrow[L\to\infty]{} 1 - c \tag{10}$$

Hence the weighting parameter c in eqn. (4) is directly related to the asymptotic adjustment of estimations.

2.4. **Mixing function**

Despite an expected value estimation, the second central component of a CM forecast model is the mixing function, see eqn. (2). Adaption to the processed data can be realized similar to eqn. (5).

$$\min \frac{1}{2}\sum_{l=1}^{L} c_l(x_l - \hat{x}_L)^2 \tag{11}$$

The prediction $\hat{x}_L$ equals the mixing function from eqn. (2). In general the mixed prediction can be a nonlinear function of its inputs, but here a linear, weighted average

$$\hat{x}_L = \sum_{i=1}^{M} w_L^i\, \hat{x}_L^i \tag{12}$$

is chosen for several reasons:

- Overfitting is hardly possible [1],

- optimal weights can be calculated,
- nonlinear relations are handled via contexts,
- its successful use under similar circumstances is reported in literature [6].

Deriving eqn. (11) and finding its roots with respect to w_L^i yields a linear equation system. The i^{th} row in the processing step L of the equation system is

$$\sum_{j=1}^{M} \underbrace{\left(\sum_{l=1}^{L} c_l \hat{x}_l^j \hat{x}_l^i \right)}_{a_L^{ij}} w_L^j = \underbrace{\sum_{l=1}^{L} c_l x_l \hat{x}_l^i}_{b_L^i} \tag{13}$$

or expressed in matrix notation

$$A_L w_L = b_L \tag{14}$$

with $A_L \in \mathbb{R}^{M \times M}, w_L, b_L \in \mathbb{R}^M$. Note that A_L is symmetric, i.e. $a_L^{ij} \equiv a_L^{ji}$. As a consequence of eqns. (12) and (13) the coefficients a_L^{ij} and b_L^i can be calculated recursively similar to eqn. (7).

$$\begin{aligned} a_L^{ij} &= \hat{x}_{L-1}^j \hat{x}_{L-1}^i + c a_{L-1}^{ij} \\ b_L^i &= x_{L-1} \hat{x}_{L-1}^i + c b_{L-1}^i \end{aligned} \tag{15}$$

Since eqn. (11) is quadratic there is just a single optimal solution for w_L per step L. There are no constrains concerning w_L, since these worsen the cost function value.

Various methods can be used to solve eqn. (14). A Conjugate Gradient approach [7] has been used successfully for this purpose.

3. LOAD DATA ANALYSIS

3.1. About the data

The present load data time series L_k is customer load obtained from a regional Thuringian (federal state of Germany) power supplier. It spans across three years, ranging from 1. Jan. 1998, 0:15 to 31. Dec. 2000, 24:00. Sampling takes place every 15 min, thus there are $H = 96$ values per day. Normalization transformes L_k into $x_k \in [0,1]$. $\underline{L}$ and $\overline{L}$ name the lowest and highest observed load value.

$$x_k = \frac{L_k - \underline{L}}{\overline{L} - \underline{L}} \tag{16}$$

3.2. Autocorrelation

The two week time series extract displayed in fig. 1 shows a daily and weekly cycle, which coincides with the autocorrelation plot in fig. 2. Typically a day is highly correlated with the same day of the previous week. More recent observations are of higher linear dependence, since the autocorrelation decreases. Therefore the load patterns change over time.

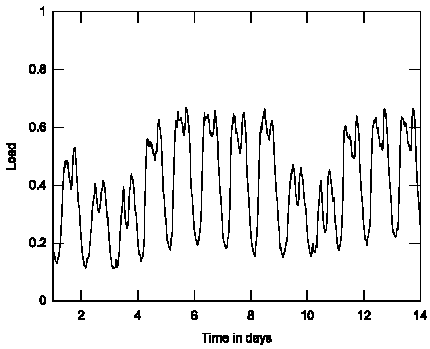

Figure 1: Normalized load data from 1. Jan 1998 to 13. Jan 1998.

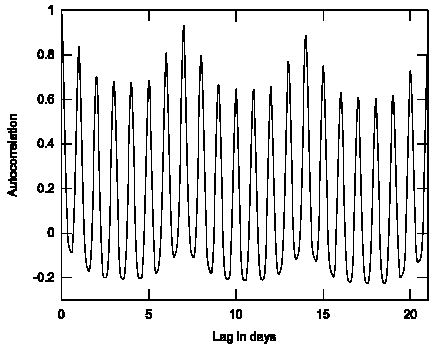

Figure 2: Load time series autocorrelation computed for 1. Jan 1998, 0:15 to 31. Dec 1998, 24:00.

3.3. Data clustering

To investigate similarities between load profiles the data is treated as a set of feature vectors $v_k \in \mathbb{R}^H$. Each vector corresponds to the k^{th} daily load pattern (0:15 to 24:00). The Fuzzy C-Means Algorithm [8] is utilized to cluster the data, which results in a set of membership degrees $\mu_j: \mathbb{R}^H \to [0,1] \subset \mathbb{R}$. Each $\mu_j(v_k)$, $1 \le j \le C$, is a measure of similarity between the j^{th} clusters' representative $\overline{v}_j$ and a load pattern v_k. In order to ease visualization each vector receives a crisp cluster assignment.

$$c_k = \arg \max_{1 \le j \le C} \mu_j(v_k) \tag{17}$$

The cluster indices j are ordered by their daily average load $E\{\overline{v}_j\}$, thus $E\{\overline{v}_i\} < E\{\overline{v}_j\}, i < j$.

$$E\{v\} = \frac{1}{H} \sum_{i=1}^{H} v_i \tag{18}$$

Fig. 3 shows the result of the clustering process. Solid vertical lines denote the time change (hence these enclose the summertime). Each point represents a load pattern according to table 1.

Marker	Small dark Circle	Plus	Big light Circle	Cross
Day	Weekday	Saturday	Sunday	Holiday

Table 1: Graphical representation of day types.

In general free days and weekdays show different behavior. Free days can further be subdivided into Saturdays, Sundays and holidays. Typically free days have a lower average load than weekdays, probably due to lower industrial activity. Sundays show the lowest electrical load. The influence of the time change is obvious – almost no clusters cross the marked dates.

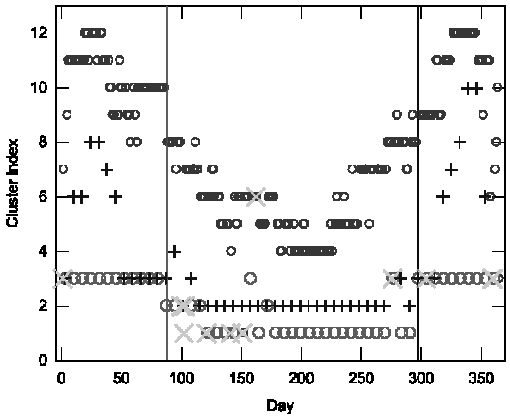

Figure 3: Clustered load profiles for 1. Jan 1998 to 31.Dec 1998, 12 clusters.

The yearly fluctuation of the average daily load (high load during cold seasons, low load during warm seasons; possibly caused by lighting and heating/cooling) can clearly be observed in fig. 3, i.e. it greatly influences the clustering process. To mask out this effect the average daily load component of a vector v is removed yielding

$$v' = (v_1 - E\{v\}\ v_2 - E\{v\} \dots v_H - E\{v\})^T \quad (19)$$

and the clustering process is repeated (see fig. 4). The segmentation of free days is more pronounced in this case. Sundays and holidays are mostly grouped into the same clusters and are more strictly distinguished from Saturdays. Still the assignment of clusters is shaped by the temporal progression, which is more evidence for the time varying characteristics of the load patterns.

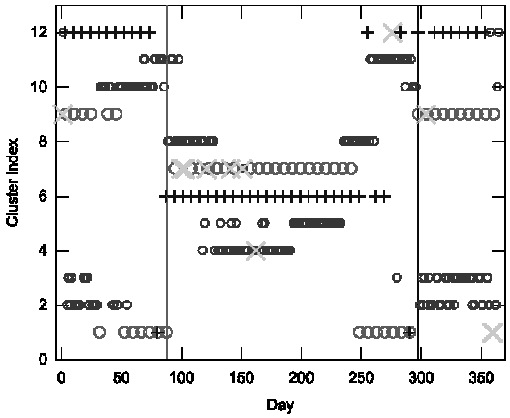

Figure 4: Clustered load profiles for 1. Jan 1998 to 31.Dec 1998, daily mean removed, 12 clusters.

4. LOAD FORECASTING MODEL

4.1. **Requirements**

Short term load forecast denotes time spans ranging from a few minutes to several days. In this case study, the forecasting horizon H is assumed to be a single day, thus $H = 96$ discrete values. $\hat{x}_{k,h}$ names the predicted value for the point h within the forecast horizon of day k.

4.2. **Context models**

A single expected value estimator from eqn. (6) used as a (sub-) model will hardly achieve good prediction performance. Hence – in analogy to data compression – a discrete context should be used to increase prediction accuracy via specialization. Context layout greatly depends on process characteristics and requirements, i.e. in particular:

- Each point $1 \leq h \leq H$ requires an estimation,
- similar day types can be clustered (e.g. holiday and Sunday) and different types should be treated separately,
- weekly periodicity should be handled.

To identify day types a bit vector $D_k = (d_{k-1} d_k d_{k+1}) \in \{0,1\}^3$ is assigned to each day k. A bit d_k identifies whether or not the day k is free. Data analysis has shown that Sunday ($D_k = 110$) and holidays (e.g. $D_k = 010$) are similar and thus clustering the observations might be beneficial. Mapping both day types to the same value of $D_k = 010$ requires to zero d_{k-1}. To formalize this procedure each possible set bit is replaced by a binary degree of freedom, $m_i \in \{0,1\}$. Table 2 shows all eight possible combinations of D_k and the resulting 12 parameters $m = (m_1\, m_2 \dots m_{12})$. $Z(D_k)$ assigns the corresponding decimal value to a bit string D_k, e.g. $Z(101) = 5$. In Table 2 $Z(D_k)$ is given for $m_i = 1$, $1 \leq i \leq 12$. The calculation of a discrete context following this technique is defined via the operator $\otimes: \{0,1\}^3 \times \{0,1\}^{12} \to [0,7] \subset \mathbb{N}$. First D_k is altered based on m, as described above. Afterwards the resulting bit string is mapped to a discrete value using $Z(\cdot)$.

$Z(D_k)$	0	1	2	3	4	5	6	7
d_{k-1}	0	0	0	0	m_5	m_7	m_9	m_{12}
d_k	0	0	m_2	m_4	0	0	m_8	m_{11}
d_{k+1}	0	m_1	0	m_3	0	m_6	0	m_{10}

Table 2: Parametrized scheme for all possible values of D_k.

The periodicity can be handled analogous. Each day k receives a day index $1 \leq w_k \leq 7$. Seven degrees of freedom, $n = (n_1 n_2 \dots n_7) \in \{0,1\}^7$, corresponding to Monday (n_1), Tuesday (n_2), … Sunday (n_7) are introduced. In a processing step k, the current day maps to its day index w_k, if $n_i = 1$. Otherwise a day index of zero is assigned. To express

this, an operator $\otimes: \{1, 2, \dots, 7\} \times \{0,1\}^7 \to [0, 7] \subset \mathbb{N}$

$$w_k \otimes n = w_k n_{w_k} \tag{20}$$

is introduced.

In the k^{th} step each point h within the prediction horizon receives a discrete context $C_{k,h}$ which quantizes the calendar information. The quantization is controlled by the parameters m and n.

$$C_{k,h} = H[2^3(w_k \otimes n) + (D_k \otimes m)] + h \tag{21}$$

4.3. **Model components**

Combining a context quantization, eqn. (21), and an expected value estimator, eqn. (6), results in a conditional expected value estimator.

$$\hat{x}_{k,h} = E\{x_{k,h} | C_{k,h}\} \tag{22}$$

The separation of steady and alternating components per load pattern can improve the discriminability. An additional parameter $b \in \{0, 1\}$ controls, if a model predicts $x_{k,h}$ or only the alternating component $x_{k,h} - \bar{x}_k, 1 \le h \le H$. Due to the separation a model for predicting the steady component $\bar{x}_k$ is required. To ease prediction $\bar{x}_k$ is decomposed.

$$\bar{x}_k = \bar{x}_{k-1} + \Delta\bar{x}_k \tag{23}$$

Since in the k^{th} prediction step, the previous steps information is known, only the estimation of $\Delta\bar{x}_k$ is required. A conditional expected value estimator from eqn. (22) can be used for this purpose.

Additional environmental factors (e.g. temperature, solar radiation, etc.) influence load patterns. This information isn't available for the given data set. However, causality can be assumed, i.e. similar environmental characteristics will result in similar load patterns. When forecasting $x_{k,h}$ (or $x_{k,h} - \bar{x}_k$) related situations $x_{j,h},\ k - W < j < k$ are identified by comparing the L latest known load samples $x_{k-1,H-h}$ to situations in the past $x_{j-1,H-h}\ 0 \le h < L$. The W most recently observed load patterns are considered.

$$\delta_{jk} = \frac{1}{L}\sum_{h=0}^{L-1}\left(x_{j-1,H-h} - x_{k-1,H-h}\right)^2 \tag{24}$$

names the similarity measure between the days j and k. Eqn. (22) is extended to yield

$$\hat{x}_{k,h} = E\left\{x_{k,h} \middle| C_{k,h} \equiv C_{j,h}\ \delta_{jk} < R\right\}. \tag{25}$$

Note that the sequence of utilized samples must be sorted descendently by δ_{jk} to assign higher weights to more similar observations, according to eqns. (4) and (6). Such an estimator adds two more degrees of freedom – the similarity radius R and the length L of the compared signal subsequence. The parameter W can be chosen based on computational resources.

4.4. **Forecasting system**

The set of forecasting submodels consists of 7 components:

- Two expected value estimator, eqn. (22),
- an extended estimator, eqn. (25),
- a single estimator $E\{\Delta\bar{x}_k | C_{k,1}\}$ and $\bar{x}_{k-1}$ according to eqn. (23),
- the most recently known point $x_{k-1,H}$ and
- a constant input $\hat{x}_{k,h} = 1$.

Individual predictions are combined using the mixing function described in section 2.4.

$$\hat{x}^8_{k,h} = m(\hat{x}^1_{k,h}, \hat{x}^2_{k,h}, \dots, \hat{x}^7_{k,h}) \tag{26}$$

A symmetric, static weighting of adjacent estimations $\hat{x}_{k,h-1},\ \hat{x}_{k,h},\ \hat{x}_{k,h+1}$ yields the final forecast.

$$\hat{x}^9_{k,h} = w\hat{x}^8_{k,h-1} + (1 - 2w)\hat{x}^8_{k,h} + w\hat{x}^8_{k,h+1} \tag{27}$$

Eqn. (27) advantage of neighboring observations' correlation. There is a single parameter, $w \in [0, 0.5] \subset \mathbb{R}$.

The forecasting system contains a rich set of parameters, including integer and real values and bitmasks. To solve the model parameter estimation problem DCGA, a Niching Genetic Algorithm [9], is used.

5. SIMULATION AND EVALUATION

5.1. **Way of processing**

The three years of present data is subdivided into training data for model fitting and testing data for evaluation. The training data spans across two years, from 1. Jan. 1998, 0:15 to 31. Dec. 1999, 24:00 and the remaining year, from 1. Jan. 2000, 0:15 to 31. Dec. 2000, serves as testing data.

To simplify the evaluation process the obtained forecasts are compared to a trivial predictor,

$$\hat{x}^T_{k,h} = x_{k-7,h}, \tag{28}$$

which takes advantage of the weekly periodicity shown in fig. 2. In the following subsection the presented models' prediction is named $\hat{x}^{CM}_{k,h}$, the same terminology is used for the prediction errors, $e^T_{k,h}$ and $e^{CM}_{k,h}$. The prediction error is given by

$$e_{k,h} = \hat{x}_{k,h} - x_{Hk+h}. \tag{29}$$

5.2. **Simulation and results**

Forecasting accuracy is measured by analyzing the statistical properties of the prediction errors. A rough overview is presented in table 3. Compared to the

naive forecast the developed model halved the magnitudes of standard deviation and minimal and maximal prediction error.

	Mean	Std.	Min.	Max.
$e_{k,h}^{T}$	$3.871\ 10^{-4}$	$4.764\ 10^{-2}$	$-3.440\ 10^{-1}$	$3.667\ 10^{-1}$
$e_{k,h}^{CM}$	$-8.506\ 10^{-5}$	$2.408\ 10^{-2}$	$-1.841\ 10^{-1}$	$1.337\ 10^{-1}$

Table 3: Characteristic values for the forecast error.

The mean is an order of magnitude lower and very close to zero.

A rough evaluation already justifies the increased computational effort of $\hat{x}_{k,h}^{CM}$ compared to $\hat{x}_{k,h}^{T}$. To judge about the average performance throughout the prediction horizon H figures 5 and 6 illustrate the mean and standard deviation of $e_{k,h}^{T}$ and $e_{k,h}^{CM}$. In both cases the mean is close to zero, $e_{k,h}^{CM}$ shows a slightly more varying picture.

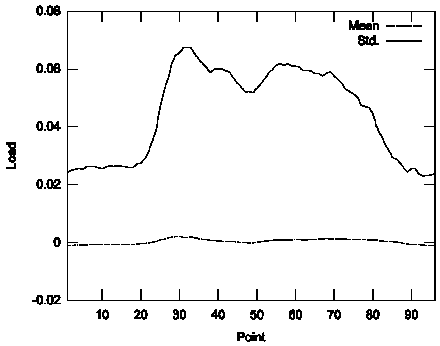

Figure 5: Mean and standard deviation (Std.) of $e_{k,h}^{T}$ from 1. Jan. 2000 to 31. Dec. 2000.

A comparsion of fig. 5 and fig. 6 shows that the standard deviation is reduced in its span and average. During the time of social activity (6:00 – 20:00) the load seems to be more difficult to predict. In fig. 5 the standard deviation reaches two plateaus (6:30 and 12:30 – 17:00). Such a behavior is suppressed by CM forecasting, i.e. much more precise forecasts can be obtained for the corresponding time range.

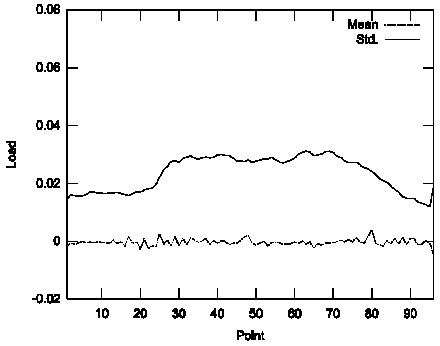

Figure 6: Mean and standard deviation of $e_{k,h}^{CM}$ from 1. Jan. 2000 to 31. Dec. 2000.

6. SUMMARY

This paper presented the paradigms of CM and derived modifications of model components to make these suitable for time series forecast. Short term load forecast served as a test case for model design. Hence a sample of load data has been analyzed to identify and take advantage of its typical characteristics. The forecasting system was constructed based on the analysis' results and fitted to the training data. An evaluation of the prediction error on the testing data showed good forecast results and a significant enhancement compared to a naive forecast.

7. REFERENCES

[1] V.H. Ferreira, A.P. Alves da Silva, "Toward Estimating Autonomous Neural Network-Based Electric Load Forecasters", IEEE Transactions on Power Systems, pp. 1554 – 1562, 2007.

[2] M. Mahoney, "Adaptive Weighing of Context Models for Lossless Data Compression", Florida Institute of Technology, Melbourne, Florida, 2005.

[3] M. Mahoney, "The PAQ1 Data Compression Program", Florida Institute of Technology, Melbourne, Florida, 2002.

[4] M. Mahoney, "Data Compression Programs", http://www.mattmahoney.net/dc/, visited Apr. 19., 2010.

[5] J. Cleary, I. Witten, "Data Compression Using Adaptive Coding and Partial String Matching", IEEE Transactions on Communications, pp. 396-402, 1984.

[6] P. Subbaraj, V. Rajasekaran, "Short Term Hourly Load Forecasting Using Combined Artificial Neural Networks", International Conference on Computational Intelligence and Multimedia Applications, Tamil Nadu, pp. 155-163, 2007.

[7] M. Hestens, E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems", Journal of Research of the National Bureau of Standards, pp. 409-436, 1952.

[8] F. Höppner, "Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition", John Wiley, Chichester, 1999.

[9] S. Mahfoud, "Niching Methods for Genetic Algorithms", University of Illinois, Urbana, 1995.