Lars Dannecker, Matthias Böhm, Wolfgang Lehner, Gregor Hackenbroich

Forcasting Evolving Time Series of Energy Demand and Supply

Erstveröffentlichung in / First published in:

Advances in Databases and Information Systems: 15th International Conference. Wien, 20.-23.09.2011. Springer, S. 302-315. ISBN 978-3-642-23737-9.

DOI: http://dx.doi.org/10.1007/978-3-642-23737-9_22

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

Forcasting Evolving Time Series of Energy Demand and Supply

Lars Dannecker[1], Matthias Böhm[2],
Wolfgang Lehner[2], and Gregor Hackenbroich[1]

[1] SAP Research Dresden, Chemnitzer Str. 48, 01187 Dresden, Germany
{lars.dannecker,gregor.hackenbroich}@sap.com
[2] Technische Universität Dresden, Database Technology Group
Nöthnitzer Str. 46, 01187 Dresden, Germany
{matthias.boehm,wolfgang.lehner}@tu-dresden.de

Abstract. Real-time balancing of energy demand and supply requires accurate and efficient forecasting in order to take future consumption and production into account. These balancing capabilities are reasoned by emerging energy market developments, which also pose new challenges to forecasting in the energy domain not addressed so far: First, real-time balancing requires accurate forecasts at any point in time. Second, the hierarchical market organization motivates forecasting in a distributed system environment. In this paper, we present an approach that adapts forecasting to the hierarchical organization of today's energy markets. Furthermore, we introduce a forecasting framework, which allows efficient forecasting and forecast model maintenance of time series that evolve due to continuous streams of measurements. This framework includes model evaluation and adaptation techniques that enhance the model maintenance process by exploiting context knowledge from previous model adaptations. With this approach (1) more accurate forecasts can be produced within the same time budget, or (2) forecasts with similar accuracy can be produced in less time.

Keywords: Forecasting, Energy, Hierarchy, Parameter Estimation.

1 Introduction

The energy market is changing from a day-ahead market to a continuous, intra-day trading that allows dynamic interactions between market participants. This new, liberalized energy market in combination with emerging smart meter technologies requires fine-grained planning capabilities. Also, the integration of more renewable energy sources (RES, e.g. wind, solar) poses additional challenges. Unlike traditional energy sources, the energy production of RES cannot be exactly planned, because the supply from RES heavily depends on external factors like the weather. It is also very inefficient to store energy produced from those sources, for what reason they have to be directly used when they are available. As a result, it is necessary to balance energy demand and supply in a fine-grained manner and establish possibilities of real-time balancing [1]. In addition, real-time balancing in combination with the hierarchical organization of the energy markets with role-specific access to relevant demand and supply data lead to a distributed data management architecture and therefore a distributed usage of forecasting models.

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

Research projects like MIRABEL [2], MeRegio [3] and many more address the issues of real-time balancing and fine-grained scheduling of energy demand and supply. To do so a fundamental requirement is the availability of accurate predictions of future energy consumption and production. For this purpose we employ model-based forecast techniques, where a quantitative model is used to describe the characteristics and behavior of historic energy time series. Most forecast models involve a number of parameters, with each describing a specific aspect of the time series (e.g., seasonal patterns, energy output). The parameters are estimated on a training data set by minimizing the forecast error (i.e., difference between predicted and actual value) that is measured in terms of an error metric like (Symmetric) Mean Absolute Percentage Error [4]. The so created instances of forecast models are used to predict future values up to a defined horizon (e.g., one day). Important classes of forecast models are: autoregressive models [5], exponential smoothing models [6] and models that apply machine learning [7]. In most cases, forecast models from these classes are specifically adapted to the special characteristics of energy time series such as multi-seasonality and the dependence on exogenous factors. Thus, they produce more accurate forecasts compared to general purpose forecasting methods. However, the necessary real-time balancing capabilities pose new challenges to forecasting of energy demand and supply. Most importantly, accurate forecasts are necessary at any point in time to allow quick adaptations of the energy schedules. Fortunately, in the energy domain the current energy consumption and production can be measured constantly. This can be seen as a continuous stream of updates append to the time series in regular intervals. To ensure high accuracy in terms of exploiting this continuous feedback it is necessary to adapt the forecast model to the updates. However, the naïve adaptation strategy of re-estimating the forecast model after each update is not applicable, because typically the estimation of a forecast model is very time-consuming, as potentially a large number of parameters, spanning an exponential search space, have to be adjusted.

Our primary contribution is a forecast model maintenance framework, which continuously monitors the forecasting accuracy and exploits context knowledge from previous model adaptations and the hierarchical system to increase the model adaptation efficiency. The core idea bases on the assumption that forecast models, i.e. the values of the parameters, only gradually change over time, which typically can be observed in the energy domain. This framework enables existing forecasting models to work with evolving time series in the context of real-time energy balancing. Furthermore, we make the following more concrete contributions that also reflect the structure of this paper:

- We introduce a concept for forecasting in distributed systems in Section 2. Our approach synchronizes the forecast models instead of exchanging forecasts or measurements and therefore reduces the communication overhead.
- Subsequently, in Section 3, we present our novel maintenance approach that includes model evaluation techniques and a parameter estimation framework to ensure efficient model adaptations on single entities of the hierarchical system.
- We compare our estimation framework to other global optimization approaches in Section 4 and show the advantages of our solution.

Finally we conclude the paper with a summary and future work discussion in Section 5.

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

2 Distributed Forecasting

The European energy market is hierarchically organized. The lowest level comprises consumers and producers organized in balance groups, where a Balance Responsible Party (BRP) manages the energy consumption and production of each group. The BRPs represent the second level. The market operators that are responsible for the market balance areas represent the third level. Additional levels are possible (e.g., Neighborhood-Oriented Energy Balancing). This hierarchical organization of the energy market in combination with different roles and role-specific access to relevant demand and supply data motivates the use of a distributed data management architecture. This also requires a forecasting solution that works in this kind of architecture.

Related Work: Currently, no solution fully adapts forecasting to a distributed system architecture. Only partial aspects are solved like the distributed collection of data. Brabec et al., presented the nonlinear mixed effects model (NLME) implemented at the customer that provides information to a central system. Challa et al. described an approach for distributed sensor networks to incorporate values from different sources into one forecasting system [8]. To do so, they used a State-Space-Model based on Vector Auto Regression combined with the Interacting Multiple Model (IMM) estimation algorithm [9]. In addition, some work was done regarding hierarchical forecasting. For example the use of an AR-GARCH Model as suggested by Sohn et al. [10] or by combining hierarchically organized models using a special regression model as discussed by Hyndman et al. [11]. However, they mostly focused on aggregation strategies with regard to local settings and accuracy only rather than the distribution of the forecasting effort. In contrast to the aforementioned approaches, our solution addresses to provide efficient forecasting functionality to a distributed energy data management system.

2.1 Distributed System Architecture

A naïve approach for forecasting in a distributed system means the direct propagation of measurements or forecasts of lower level entities to the responsible next level entity. This entity then calculates a global forecast. This approach exhibits several drawbacks such as limited transmission granularity due to privacy restrictions that only allows data transmission every 15 minutes and a large communication overhead (e.g., 700 million customers in Europe sending data every 15 minutes).

For this purpose we introduce a more sophisticated forecasting approach for a distributed environment. The approach is based on the assumption that demand and supply measurements are available for all entities in the hierarchy, which motivates a more independent forecasting between hierarchy levels. Figure 1 presents an independent distributed forecasting approach that involves model synchronization. Each entity (S) calculates its own forecasting (F - forecast model) based on its own measurements. No measurements or predicted values are communicated. We rather suggest a synchronization of the forecast models between the different levels to still recognize local changes and to guarantee consistency between forecasts at different hierarchy layers. Model synchronization is conducted less frequently and by fewer entities compared to periodic measurement transmissions from all entities. Therefore, the communication effort

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

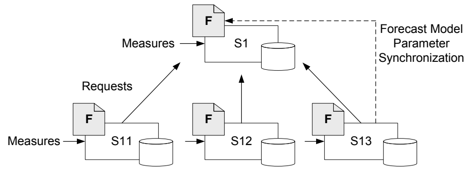


Fig. 1. Distributed Forecasting with Model Synchronization

between hierarchy levels is significantly reduced. In addition, privacy restrictions do not apply, because no measurements or predictions are transmitted. Thus, the forecasting can be based on data with a granularity lower than 15 min. Furthermore, model synchronization provides context knowledge about model adaptations on lower levels that can be used to enhance the model adaptation process on higher level entities. In the following, we describe the model synchronization in such a distributed architecture.

2.2 Model Synchronization

The goals of the model synchronization approach is to ensure the consistency of the forecasting results between the levels of the hierarchy, e.g., the aggregation of individual lower level forecasts should almost provide the same result as the forecast on the upper level. To reduce the communication efforts we assume that the impact of most single entities on the global forecasting is rather low, where the impact of an entity can be estimated by its share on the total consumption and production of its group. Therefore, it is not necessary to adapt the forecast model on higher levels for each model adaptation on a lower level entity. However, a large group of single entities could create a critical mass that is large enough to generate impact on the global forecasting. In addition, large customers, e.g., big companies, could have an impact that is sufficient to influence the forecast on upper level entities. We therefore, use a propagation strategy that involves selective change notifications from lower levels. This means that the change notification is only transmitted to the responsible next level entity. The model synchronization process works then as follows:

1. An entity adapts its forecast model (triggered by local model evaluation [as described in Section 3.2]).
2. A notification is sent to the responsible entity on the next level that includes a description of the model adaptation. The descriptions are transmitted as change vectors, containing the old and new forecast model parameters.
3. The next level entity collects the notifications until a critical mass with sufficient impact on its own forecast model is reached. The exact calculation of the critical mass is subject to future work.
4. Afterwards, an adaptation of the forecast model on the next level is performed using the change vectors of the lower level entities as input for the optimization.

This approach reduces the communication efforts by notifying responsible entities only when a forecast model adaptation is triggered. The information about the model adaptations can be used to enhance the model adaptation process on the next level entity.

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

For this purpose, before starting the model adaptation process the most recent changes on the child entities are considered, avoiding the usage of outdated information.

3 Forecast Model Maintenance

To ensure an efficient forecasting in the hierarchical system, it is important to also consider the forecasting processes on a single system entity. As mentioned before time series in the energy domain evolve over time. To guarantee up-to-date forecast models at any point in time a continuous model maintenance is needed. Also, additional energy domain specific particularities pose the challenge of causing unpredictable but gradual changes: (1) The energy consumption and the supply of renewable energy sources is influenced by uncertain exogenous factors like weather or temperature. (2) Real-time balancing capabilities allow market actors to constantly adapt energy consumption and production. These particularities require efficient model maintenance. In this section, we introduce the overall forecast model maintenance strategy for single system entities, which includes several model evaluation and adaptation techniques.

3.1 Maintenance Strategy Overview

Figure 2 illustrates our forecast model maintenance strategy that consists of three steps. First, for each new measurement we initiate an update of the local model. This step includes the incremental state adaptation (e.g., smoothing constant) of the forecast model and the persistent storage of measurements. This update is a simple insertion such that it is not the focus of this paper. Second, we continuously evaluate the accuracy of the forecast model upon evolution of the time series using different model evaluation techniques. Third, based on the outcome of the accuracy evaluation the adaptation of the forecast model to the new situation is triggered, which means a re-estimation of the parameters involved in the forecast model. When considering complex models that describe a lot of information like multiple seasons and exogenous information, the models involve a high number of parameters. Each parameter adds an additional dimension to the solution space, which increases the amount of solutions that have to be evaluated. For this reason the parameter re-estimation can be very time consuming. Triple Seasonal Holt Winters [12] for example involves five parameters, which leads to a number

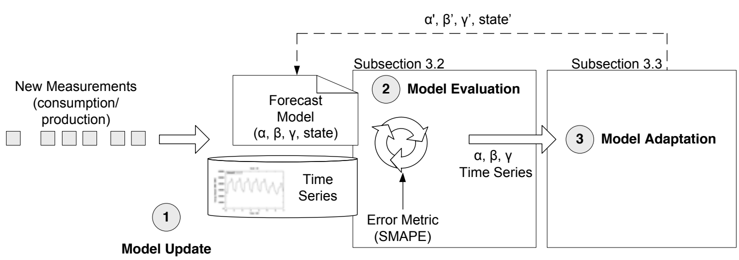


Fig. 2. Maintenance Strategy for Evolving Time Series

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

of x^5 possible parameter combinations, with x being the granularity of a parameter. The autoregressive multi-equation model EGRV [13] and its adaptations (e.g., Cottet et al. [14] and Dordonnat et al. [15]) model each hour as a separate model which leads to an even larger parameter space (e.g., EGRV: 24 times up to 31 parameters). As a result, parameter estimation approaches that efficiently find new optimal parameter combinations are needed to adapt even complex forecast models in reasonable time.

3.2 Forecast Model Evaluation

The second step in the maintenance process is the model evaluation. We distinguish two major groups of evaluation strategies: First, fixed interval model evaluation, where the model adaptation is triggered periodically. This strategy does not evaluate the forecast error. Therefore, it exhibits the problem of determining a reasonable model adaptation interval. Too short intervals mean unnecessary adaptations, whereas too long intervals pose the risk that arbitrary large errors may build up between the model adaptation intervals. The threshold-based model evaluation is the second strategy. It continuously evaluates the forecast model accuracy and triggers a model adaptation when the forecast error violates a previously defined error threshold. This enables quick adaptations of the forecast model to changes of the evolving time series. While this strategy guarantees that a certain forecast error is not exceeded, it exhibits a similar drawback like the fixed interval model adaptation as it also depends on the definition of suitable thresholds.

Due to the high influence of the adaptation criteria, we propose a heuristic approach that combines evaluation strategies. A combination weakens the disadvantages of the single techniques. There, a model adaptation is periodically triggered after a specified amount of time. Also, the model is continuously evaluated and adapted each time the error surpasses a defined threshold. The time counter is reset after a model adaptation was triggered by a threshold violation, to avoid unnecessary adaptations of the forecast model. The combination of model maintenance strategies reduces the dependence on single adaptation criteria, which makes it easier to determine suitable thresholds.

3.3 Enhanced Parameter Estimation

Once a model adaptation has been triggered, we try to re-estimate the parameters of the forecast model, which typically is a very time-consuming task. For this reason, a parameter estimation method that efficiently finds a new optimal parameter combination with regard to the forecast error is necessary.

Related Work: For the optimization of forecast model parameters several algorithms exist, which can be divided into two classes: (1) Algorithms that need a derivable function and (2) algorithms that can be used with arbitrary functions. In this paper, we only focus on the class of algorithms that do not require a derivable function, because they are more general and can be used with arbitrary forecasting models and error functions. These algorithms are classified into local and global optimization algorithms. Global optimization algorithms consider the whole solutions space with the goal of finding a solution that is the global optimum. Thus, in general they need more time to terminate.

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

In contrast, local optimization algorithms follow a directed approach and therefore converge faster to a solution but exhibit the risk of starvation in local suboptima. Also, they strongly depend on the position of a provided starting point that has to be close enough to the optimum to guarantee convergence. Examples for global optimization algorithms are: Simulated Annealing [16] or genetic algorithms [17]. Examples for local optimization are: Hook-Jeeves [18] and Nelder-Mead [19]. In addition to the described optimization algorithms, we can find parameters empirically using a naïve method called grid search that sequentially evaluates all solutions in a given granularity. However, its runtime exponentially increases with the number of parameters. Due to the limitations of the naïve method as well as local and global optimization algorithms, we enhance the parameter estimation process by introducing our parameter estimation framework.

Our core idea is to exploit context knowledge of previous model adaptations by determining starting points using information from the model synchronization within the hierarchical system and the previous parameter combination. The underlying assumption is that the combined parameter changes of the child entities approximately reflect the parameter changes of the forecast model at the parent entity. In addition, due to the continuous model evaluation we also assume that forecast model parameters will not change abruptly, but will be in the neighborhood of previous parameters. Our approach exploits both assumptions by combining the parameter changes suggested by the model synchronization with the previous parameter combination. Thus, we reduce the problem of finding a global optimum to the problem of finding local suboptima in the near surrounding of the determined starting point.

We supplement this local strategy with global coverage approaches. For this purpose, we introduce a parameter estimation framework for different search strategies that is illustrated in Figure 3: A starting point is determined by exploiting context knowledge of the continuous model maintenance and adaptation information of child entities in the hierarchical system. This starting point serves as input for a two-phase optimization process that refines the starting point to find a new, optimal parameter combination. The process consists of a local search for fast convergence and a global search to reduce the probability of getting stuck in local suboptima. In the following we describe (1) how to determine starting values for the parameter estimation process and (2) how to optimize this initial solution to find the global optimal parameter combination in detail.

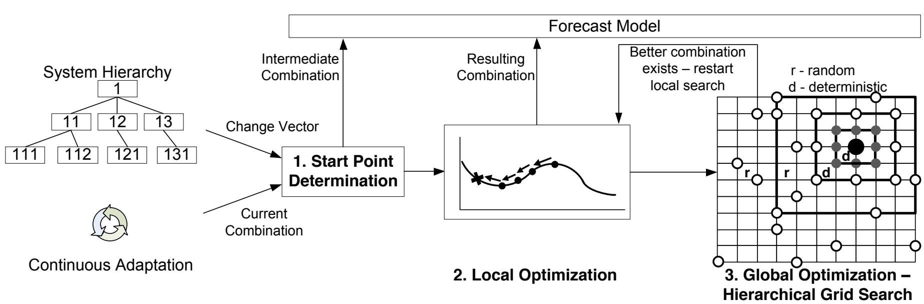


Fig. 3. Model Adaptation Process

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

Starting Point Determination. For local optimization approaches a good starting point is essential for finding a global optimal solution. Also, the runtime of such approaches is directly influenced by the position of the starting points. The closer the location of the starting point to the optimum, the lower the time needed to converge.

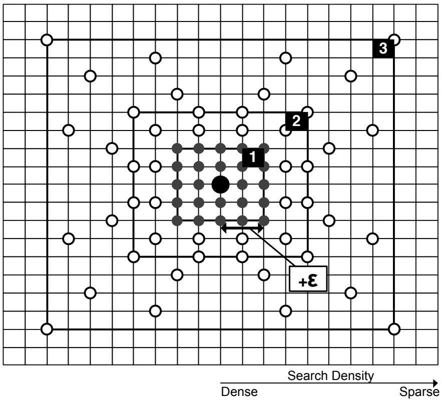
In our solution the forecast model is evaluated continuously and adapted regularly, which leads to the assumption of rather slight adaptations of the forecast model configuration. For the parameter estimation we therefore assume that the probability of finding the new global optimal parameter combination is highest in the near surrounding of the current combination. Given this assumption we set the current parameter combination as a starting point for any subsequent search strategy.

We further enhance the starting point determination by using the context knowledge of model maintenance from child entities in the hierarchy. The basis is the parameter change vector that is exchanged during model synchronization (compare Section 2.2). This vector contains the parameter combinations before and after the model adaptation of an entity, which means that the change vector represents the transition between forecast model configurations. These two parameter combinations can be used to estimate the direction of the parameter change by calculating their difference. We could also directly provide the difference of both vectors, but to allow more complex operations in the future we chose a more general solution in providing the parameter combinations. We can assume that the combination of all forecast model adaptations from entities on the lower level nearly represents the necessary changes to the model of the responsible parent entity. This is reasoned by the fact that the forecasting of the parent entity reflects the aggregated consumption or production of all connected child entities. To determine the starting point first, a global change vector is computed by combining the parameter change directions of the change vectors of multiple system nodes, weighting them according to the integral of consumed or produced energy. It is important to note that deviations from single entities are equalized due to the aggregation of many entities and the weighting. Subsequently, the new starting point is computed as the arithmetic mean of the old parameter combination and the global change vector.

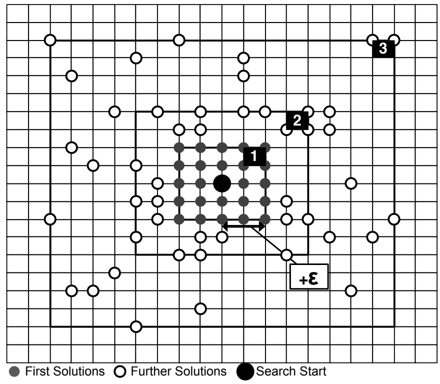
In conclusion, for entities at the lowest level we exploit the continuous model adaptation with the assumption that the new global optimal solution is located in the near surrounding of the old parameter combination. On higher hierarchy levels, we use a combination of old parameter values and propagated changes of child entities to determine the start value. As a result, we compute start values that have a high probability of being close to the global optimal solution and thus, serve as good initial values.

Optimization Process. With a good starting point in place and assuming to find the optimal solution in the near surrounding of this starting point, we define a parameter estimation framework that allows fast convergence and that ensures stochastic properties of finding the global optimal solution likewise. The process comprises of the following two phases: First, we use the determined starting point as the input for a local optimization approach like the Nelder-Mead algorithm [19]. This simplex-based algorithm iteratively evaluates the neighborhood of the starting point until a local optimum is found. Due to the starting point determination the probability of the local optimum to be also the global optimal solution is fairly high. At the same time the local optimization with given starting points converges relatively fast. After the local optimum has been found,

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22



(a) Hierarchical/Deterministic



(b) Hierarchical/Stochastic

Fig. 4. Global Coverage: Enhanced Grid Search Variations

in the second phase we try to ensure its global optimality. For this purpose we start a hierarchical optimization that converges to the global optimal solution:

1. We span an area with a user defined margin $\pm\epsilon$ in all dimensions of the solution space around the local optimum.
2. Within this area, we sequentially evaluate all possible parameter combinations, which we did not evaluate in the first phase, in a predefined granularity.
3. We double the margin ϵ spanning a second area around the first one.
4. We repeat the search within this area. There, the search granularity is increased by considering the same amount of parameter combinations as in the first iteration, but distributed over an area that is 2^d times larger than the first one.

This process is repeated until the end of the grid is reached. If a solution is found that is better than the current best one, the local search starts again from this combination. It is important to note that for different start values during the continuous model adaptation process, the parameter combinations of the outer areas evaluated by this process also change. Thus, over time, the probability to find possible optima in other regions increases. An example of the deterministic approach is illustrated in Figure 4(a).

In addition to the standard deterministic search, a second possibility is to stochastically evaluate the solution space given a specific probability density function. It is faster due to the fact that no specific order of the results is required. Also, in contrast to the deterministic search, over time the stochastic search asymptotically evaluates all possible combinations in expectation. This leads to the following search strategy: In the ϵ-area all possible solutions are considered. In the further areas solutions are picked randomly, while the number of considered solutions always corresponds to the number of solutions in the core area. This leads to a more coarse-grained search resolution. Figure 4(b) illustrates an example of the hierarchical, stochastic search.

The size of the ϵ-area and the search resolution (res) can be defined freely. Nevertheless both directly influence the number of considered solutions. When considering

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

complex forecast models (e.g., EGRV >30 parameters) the number of considered solutions in the ϵ-area increases exponentially with the number of parameters. We therefore define a maximal number of solutions in the ϵ-area, $maxPoints$, and therefore limit the maximal runtime of the enhanced grid search. To further reduce the run time of the global coverage, we can add some aspects of the well-known hill climbing optimization approach by terminating the grid search when in a subsequent area no better solutions is found than the current best. For example, if no better solution is found after the second expansion of the search area, the algorithm terminates. Essentially, the enhancements reduce the runtime, while sacrificing robustness.

4 Experimental Evaluation

In this section, we present our experiments that show the benefits of our parameter estimation framework. We demonstrate that precomputed start values enhanced by a local and a global search yield better results than solutions involving multiple random starting points. We validate our claims by comparing our parameter estimation framework to other global optimization algorithms.

4.1 Experimental Setting

We base our experiments on the publicly available data set from the UK National Grid organization. The data set contains metered electricity demand of the United Kingdom (UK) from April 1st 1971 to December 31st 2009. For our experiments we used the INDO[1] measure from January 1st 2002 to December 31st 2009 in a half hour granularity.

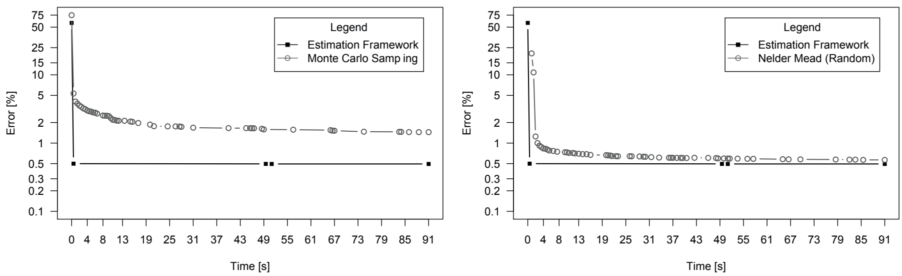
For the computation of the forecast, we chose the triple seasonal Holt Winters Exponential Smoothing (HWT). This model is tailor made for data from the energy domain and performs well on the above mentioned data set [12]. For the estimation process we split our data set as follows: The model was initialized with the years 2002 to 2008. We forecasted the year 2009 using a one-step ahead forecast to evaluate the applicability of the tested parameter combination. The forecasting error was calculated with the SMAPE error metric [4]. For our evaluation we used the following environment: Intel Core 2 Duo 2,0 GHz, 3 GB RAM, Windows 7 32bit operating system. All experiments were implemented using the C++ programming language.

4.2 Comparison of Efficiency and Accuracy

Parameter Estimation Framework versus Monte Carlo Grid Sampling: In the first experiment, we compared our parameter estimation framework to Monte Carlo grid sampling. Unfortunately, we did not have a hierarchical system with sufficient measurement data in place. For this reason, we simulated the start value for our framework by executing a coarse-grained grid search with a step size of 0.25 and subsequently modified the found point by 0.005 in random directions. The grid is configured with an ϵ-area of 0.1 and a search resolution of 0.05. Monte Carlo grid sampling chooses points from the solution space in a random fashion without using any further optimization. This method is

[1] "**INDO** - Initial Demand Outturn based on operational generation metering." [20]

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22



(a) Framework vs. Monte Carlo Sampling (b) Framework vs. Nelder Mead (Random)

Fig. 5. Experiment: Comparing Accuracy and Efficiency

much faster in selecting possible solutions but it is totally stochastic with uniform PDF. Thus, we repeated the experiment 20 times and used the average of all attempts as the result. For both solutions, we tracked the development of the best-found solution.

Figure 5(a) illustrates the result of this experiment, which shows that at no point in time the Monte Carlo grid sampling exhibited a better development than the parameter estimation framework. The error progression is always below the curve of the Monte Carlo grid sampling with a minimal forecast error of 0.494%. In comparison the minimal forecast error of the Monte Carlo grid sampling was 1.446% using the same runtime as the estimation framework. The framework found its best solution already after 51 s, while the subsequent time was used for the global coverage. The framework quickly converged to a local optimum with a forecast error of 0.499% the subsequent grid search then found a better solution (forecast error 0.495%) that was then again enhanced by the Nelder-Meads algorithm to the global optimal solution. The Monte Carlo grid search started with a higher error value and converged slower than the parameter estimation framework. This clearly shows the benefit of using optimization algorithms instead of just randomly selecting points from the grid.

The overall time to converge is rather short with 91 s, however, when considering more complex models or longer time series the time frame needed to converge to the global optimum will increase considerably. When using the EGRV model with 22 parameters (22 instead of 31 due to missing weather information), where we limited the maximal considered solutions in the ϵ-area to 1000, the parameter estimation took 635 s.

Parameter Estimation Framework versus Local Search with Random Starting Points: In the second experiment we compared our parameter estimation framework to a repeated local optimization with random starting points. Since the local search is repeated with multiple starting points, it is equivalent to a global optimization. To ensure comparability we also chose Nelder-Mead for local optimization. Our estimation framework used the same configuration as in the first experiment.

The results are illustrated in Figure 5(b). For the local search with random starting point we exhibit a better error development compared to the Monte Carlo grid sampling. However, it still converged slower and did not reach an error as low as our estimation framework. The average error value of the local search with random starting point after 91 s was 0.566%. At any point in time we get a better solution with our parameter

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

estimation framework. To find out the timeframe necessary for the local search with random starting point to converge to the same result as the parameter estimation framework, we removed the time limitation. The overall runtime was ten hours. The best results was found after 100 min with an error value of 0.499%. This also suggests that no better solution exists in the search space, which means that our estimation framework found the global optimal solution. In conclusion, the computation of a decent starting point in combination with our estimation framework is beneficial compared to just randomly selecting points as an input for the local optimization. We also showed that the subsequent global optimization is necessary to ensure that the global optimal solution is found.

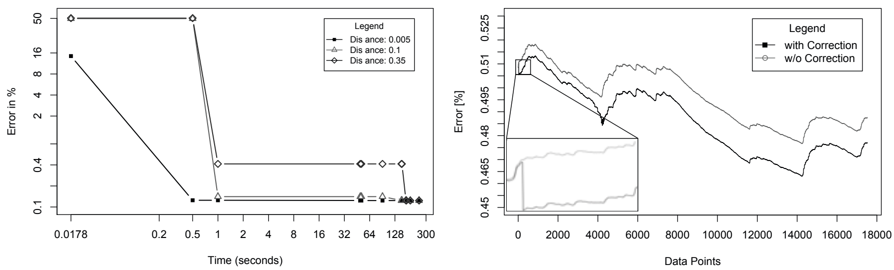
Parameter Estimation Framework versus Nelder-Mead with Simulated Annealing: In this experiment, we compare the error development of our framework using enhanced grid search and simulated annealing for global coverage. The results show that first, both algorithms converge to the same intermediate solution of 0.499%, because both use the same local search algorithm. However, subsequently the enhanced grid search found a better solution that was again refined by the Nelder-Mead algorithm to 0.494%. In contrast, the simulated annealing algorithm did not find a better solution. An explanation is that our enhanced grid search considers solutions with minimal granularity in the near surrounding of the starting point, while the simulated annealing randomly selects points from the whole solution space.

Increasing Distance to the Starting Point: The starting point used for our framework was pre-computed via a coarse-grained grid search to simulate that the starting point is the last know parameter combination. To estimate the dependence of our framework to the starting point, in this experiment we increased the distance to the starting point from 0.005 to 0.1 and 0.35. The results are illustrated in Figure 6(a). They show that in all cases a similar error value is reached, but with a different convergence speed. For a distance of 0.1 the framework first converged slower to an intermediate solution that is tight above the result when using a distance of 0.005. The global optimal solution was reached after 194 s compared to 91 s. Using a distance of 0.35 the global optimal solution was not reached. The framework found another solution with an error value of 0.4943% that is almost as good as the global optimal solution with 0.4942%. As a result, when increasing the distance the timeframe to converge to the global optimal solution is increased and the chance of finding the global optimum is reduced.

Simulation of a Evolving Time Series: In this experiment, we simulated an evolving time series to demonstrate the knowledge exploitation of previous model adaptations. We used the data from the years 2002 to 2007 to initialize the model, the year 2008 to evaluate the initial parameters and the year 2009 to simulate the evolving time series. We continuously monitored the error and triggered a model adaptation when either the error threshold was violated (defined: 0.5% SMAPE) or once a day (48 data points). Our estimation framework was used with an ϵ-area of 0.1 and a search resolution of 0.1.

Figure 6(b) illustrates the error development over the year 2009. We observed the typical behavior of higher error values in the winter months, because there are more peak demands in the winter due to illumination and heating than in the summer. These peaks are harder to predict. Using a continuous evaluation and model adaptation, the error is always lower compared to a solution without continuous maintenance. This

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22



(a) Scalability: Distance to Starting Point (b) Continuous Model Maintenance

Fig. 6. Experiments: Continuous Model Maintenance & Distance to Starting Point

shows the benefit of continuously adapting the forecast model to the current situation. In our evaluation the model adaptation based on previous parameters needed 28.23 s on average. The model adaptation was triggered 388 times and a better model was found 183 times. The small extract in Figure 6(b) illustrates the effect of a single parameter re-estimation. There, the error is reduced from 0.510% to 0.505%. In conclusion, our concept of exploiting knowledge from previous model adaptation is beneficial, because our framework finds parameter combinations that improve the forecast accuracy.

5 Conclusion

The liberalized energy market requires the balancing of energy demand and supply in real-time. This poses the challenge of forecasting in a distributed environment and the need for continuous model maintenance to provide reliable forecasts. In this paper, we introduced a forecasting approach for a hierarchical energy management system. This approach uses model synchronization instead of communicating measurements or forecasts to reduce the communication overhead. Thus, it allows an efficient forecasting in a distributed environment. We then investigated the maintenance of forecast models on a single entity and introduced our estimation framework that exploits context information from previous model adaptations and the hierarchical system to compute a suitable starting point for further optimization. The framework employs a combination of local and global search algorithms to find the global optimal solution in reasonable time, whereas arbitrary local and global algorithms can be used. Our evaluation shows the benefits of precomputed start values in combination with our estimation framework to increase the probability of finding the global optimal solution. In conclusion, we presented an efficient way to apply forecasting to distributed environments and to continuously maintain the forecast models on the entities of the system. With our approach efficient forecasting in distributed environments is possible. In addition, more accurate forecasts can be produced within the same time budget, or forecasts with similar accuracy can be produced in less time. In the future we will extensively evaluate the model synchronization approach to estimate its concrete potential in the application context.

Acknowledgment. The work presented in this paper has been carried out in the MIRACLE project funded by the EU under the grant agreement number 248195.

Final edited form was published in "Advances in Databases and Information Systems: 15th International Conference. Wien 2011", S. 302-315, ISBN 978-3-642-23737-9
http://dx.doi.org/10.1007/978-3-642-23737-9_22

References

1. Belhomme, R., Asua, R.C.R.D., Valtorta, G., Paice, A., Bouffard, F., Rooth, R., Losi, A.: Address - active demand for the smart grids of the future. In: CIRED Seminar 2008: SmartGrids for Distribution (2010)
2. Project, M.: (2011), http://www.miracle-project.eu
3. MeRegio Project (2011), http://www.meregio.de/en/
4. Hyndman, R.J.: Another look at forecast-accuracy metrics for intermittent demand. Foresight: The International Journal of Applied Forecasting 4, 43–46 (2006)
5. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control. John Wiley & Sons Inc., Chichester (2008)
6. Winters, P.R.: Forecasting sales by exponentially weighted moving averages. Management Science, 324–342 (April 1960)
7. Bunnoon, P., Chalermyanont, K., Limsakul, C.: A computing model of artificial intelligent approaches to mid-term load forecasting: a state-of-the-art- survey for the researcher. Int. Journal of Engineering and Technology 2(1), 94–100 (2010)
8. Challa, B., Challa, S., Chakravorty, R., Deshpande, S., Sharma, D.: A novel approach for electrical load forecasting using distributed sensor networks. In: Proceedings of the Third ICISIP, pp. 189–194 (2005)
9. Blom, H., Bar-Shalom, Y.: The interacting multiple model algorithm for systems with markovian switching coefficients. IEEE Transaction on Automatic Control 33, 780–783 (1988)
10. Sohn, S.Y., Lim, M.: Hierarchical forecasting based on ar-garch model in a coherent structure. European Journal of Operational Research 176, 1033–1040 (2005)
11. Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G.: Optimal combination forecasts for hierarchical time series. Technical report, Monash University (2007)
12. Taylor, J.W.: Triple seasonal methods for short-term electricity demand forecasting. European Journal of Operational Research 204, 139–152 (2009)
13. Ramanathan, R., Engle, R., Granger, C.W., Vahid-Araghi, F., Brace, C.: Short-run forecasts of electricity loads and peaks. International Journal of Forecasting 13, 161–174 (1997)
14. Cottet, R., Smith, M.: Bayesian modeling and forecasting of intraday electricity load. Journal of the American Statistical Association 98, 839–849 (2003)
15. Dordonnat, V., Koopman, S., Ooms, M., Dessertaine, A., Collet, J.: An hourly periodic state space model for modelling french national electricity load. International Journal of Forecasting 24(4), 566–587 (2008)
16. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science New Series 220(4598), 671–680 (1983)
17. Fogel, D.B.: Evolutionary Computation: Towards a New Philosophy of Machine Intelligence. IEEE Press, Los Alamitos (2000)
18. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. Journal of the ACM 8, 212–229 (1961)
19. Nelder, J., Mead, R.: A simplex method for function minimization. The Computer Journal 13, 308–313 (1965)
20. nationalgrid: nationalgrid UK - Metered half-hourly electricity demands (2010), http://www.nationalgrid.com/uk/Electricity/Data/Demand+Data/