Electricity Load Forecast using Data Streams Techniques

Pedro Pereira Rodrigues and João Gama

LIACC-NIAAD, University of Porto
R. de Ceuta 118-6
4050-190 Porto,Portugal
{prodrigues,jgama}@liacc.up.pt

Abstract. Sensors distributed all around electrical-power distribution networks produce streams of data at high-speed. From a data mining perspective, this sensor network problem is characterized by a large number of variables (sensors), producing a continuous flow of data, in a dynamic non-stationary environment. In this work we analize the most relevant data mining problems and issues: online learning and change detection. We propose an architecture based on an online clustering algorithm where each cluster (group of sensors with high correlation) contains a neural-network based predictive model. The goal is to maintain in real-time a clustering model and a predictive model able to incorporate new information at the speed data arrives, detecting changes and adapting the decision models to the most recent information. We present preliminary results illustrating the advantages of the proposed architecture.

Keywords: Electricity demand forecast, online clustering, incremental neural networks.

1 Motivation

Electricity distribution companies usually set their management operators on SCADA/DMS products (Supervisory Control and Data Acquisition / Distribution Management Systems). One of their important tasks is to forecast the electrical load (electricity demand) for a given sub-network of consumers. Load forecast is a relevant auxiliary tool for operational management of an electricity distribution network, since it enables the identification of critical points in load evolution, allowing necessary corrections within available time. In SCADA/DMS systems, the load forecast functionality has to estimate, on a hourly basis, and for a near future, certain types of measures which are representative of system's load: active power, reactive power and current intensity. In the context of load forecast, near future is usually defined in the range of next hours to the limit of seven days, for what is called *short-term* load forecast.

Traditionally, real knowledge extraction problems faced a barrier on the relative scarcity of data. Nowadays, not rarely the amount of available data is so huge that traditional systems, based on memory and several reading of same information, cannot operate efficiently. Moreover, on current real applications, data are being produced in a continuous flow, at high speed, producing examples over time [4]. In this context, faster answers are usually required, keeping an anytime model of the data, enabling better decisions.

Given its practical application and strong financial implications, electricity load forecast has been targeted by innumerous works, mainly relying on the non-linearity and generalizing capacities of neural networks, which combine a cyclic factor and an auto-regressive one to achieve good results [5]. Nevertheless, static iteration-based training, usually applied to estimate the best weights for network connections, is not adequate for the high speed production of data usually encountered. Moreover, a predictive system may be developed to serve a set of thousands of load sensors, but the load demand values tend to follow a restrict number of profiles, considerably smaller than the total of registered sensors. This way, clustering of sensors greatly allows the reduction of necessary predictive models. However, most work in data stream clustering has been concentrated on example clustering and less on variable clustering [8].

2 General Description

The main objective of this work is to present an incremental system to predict in real time the electricity load demand, in huge sensor networks. The system must predict the value of each individual sensor with a given temporal horizon, that is, if at moment t_i we receive an observation of all network sensors, the system must execute a prediction for the value of each variable (sensor) for the moment t_i+k. In this scenario, each variable is a time series and each new example included in the system is the value of one observation of all time series for a given moment.

Given the high dimensionality of the problem, a scheme that would generate a predictive model for each variable is not possible. Our approach is to first cluster the sensors using an online data stream clustering algorithm, and then associate to each cluster a neural network trained incrementally with the centroid of the cluster.

The system applies a divisive strategy, with the leaves representing the sensor clusters, which may aggregate in case of changes in the correlation structure. Anytime a cluster is divided, the offspring leaves inherit the parent's predictive model, starting to train a different copy.

The neural networks are trained continuously with data from the corresponding variables, assuming a model for each group since it is expectable that clustered variables should behave with high correlation. Nevertheless, the system makes predictions for all variables, independently, in real time. Overall, the system predicts all variables in real time, with incremental training of neural networks and continuous monitoring of the clustering structure.

3 Incremental Clustering of Data Streams

The clustering strategy used in this system is the ODAC (Online Divisive-Agglomerative Clustering), a variable clustering algorithm that creates a hierarchical structure of clusters in a divisive way [8]. The leaves are the resulting clusters, with a set of variables in each leaf. The union of the leaves is the complete set. The intersection of leaves is the empty set.

The system includes an incremental dissimilarity measure based on the correlation between time-series. The similarity measure is calculated with sufficient statistics gathered continuously over time. There are to main operations in the hierarchical structure of clusters: *expansion* that splits one cluster into two new clusters, and *aggregation* that aggregates two clusters into one. Both operators are based on the diameters of the clusters, and supported by confidence levels given by the Hoeffding bounds [6]. The main characteristic of the system is the monitoring of those diameters.

In a hierarchical structure of clusters, created from a divisive point of view, and considering that the data streams are produced by a stable concept, the intra-cluster dissimilarity should decrease with each split, therefore diminishing the diameter of each cluster. For each cluster, the system chooses two variables that define the diameter of that cluster (those that are less correlated). If a given heuristic condition is met on this diameter, the system splits the cluster in two, assigning each of those variables to one of the two new clusters. Afterwards, the remaining variables are assigned to the cluster that has the closest pivot (first assigned variables). The newly created leaves start new statistics, assuming that only the future information will be useful to decide if the cluster should be split. This characteristic increases the system's ability to react to changes in the concept as, later on, a test is executed so that if the diameter of the leaves approaches the diameter of the parent node, then the previous split may have ceased to represent the current structure of data. When this happens, the system aggregates the leaves, restarting the sufficient statistics for that group.

The ODAC algorithm presents the needed characteristics for adaptive learning systems, namely the ones related with incremental clustering [2]. Moreover, it is one of the first systems clearly proposed to hierarchical clustering of variables over data streams. An important characteristic of this method is that each split of a leaf with n variables reduces the global number of dissimilarities to compute, by at least $n-1$. The temporal complexity of each iteration of the clustering system is constant given the number of examples, even decreasing whenever a split occurs. This way, it is able to process high speed data streams. Figure 1 presents the resulting hierarchy of the clustering procedure.

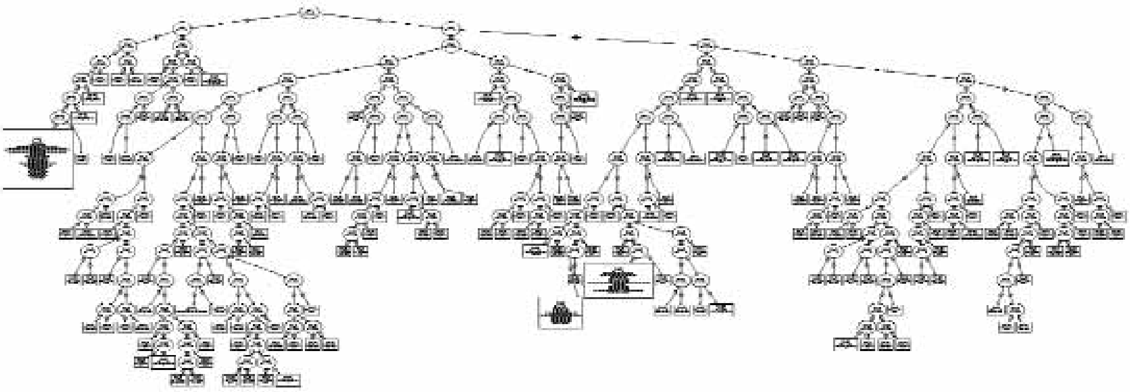

Fig. 1. ODAC clustering hierarchy in the Portuguese Electrical Network (2500 sensors in one year data).

4 Incremental Learning of Neural Networks

A neural network model is associated with each group defined in the clustering structure, in order to perform predictions on the corresponding variables. On a first approach, we focus on predicting the next hour load. At each moment t_i, the system executes two actions: one is to predict the moment t_{i+k}; the other is to back-propagate in the model the error, obtained by comparing the current real value with the prediction made at time t_{i-k}. The error is back-propagated through the network only once, allowing the ability to cope with high speed streams. Although the system builds the learning model with the centroid of the group, the prediction is made for each variable independently. Every time a cluster is split, the offspring clusters inherit the parent's model, starting to fit a different copy separately. This way, a specification of the model is enabled, following the specification of the clustering structure. When an aggregation occurs, due to changes in the clustering structure, the new leaf starts a new predictive model. From the user's point of view, there is now need to predict variables which do not behave accordingly to what is expected. Therefore, our system will only start learning a neural network if and when the corresponding cluster presents good intra-cluster correlation. This heuristic supports the notion that the majority of variables in a group present positive correlation with each other. For groups where no predictive model was created, the prediction is to consider the last known value of the corresponding stream.

The chosen predictive model was the feed-forward neural networks, with 10 inputs, 4 hidden neurons (tanh-activated) and a linear output. The input vector for predicting time series t at time k is k minus $\{1, 2, 3, 4\}$ hours and k minus $\{7, 14\}$ days. As usual [5], we consider also 4 cyclic variables, for hourly and weekly periods (*sin* and *cos*). The initial learning algorithm was the *iRprop*, reducing the system's sensitivity to parameters [7]. To introduce predictive ability in ODAC it was necessary to consider buffering the data, so that neural networks could use historical values of each variable to predict. Figure 2 presents a general description of the procedure executed at each new example.

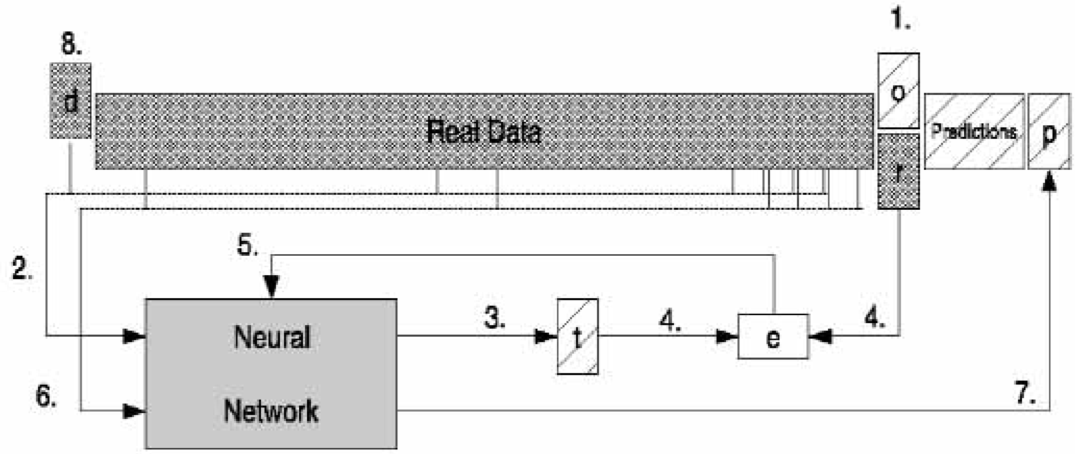

Fig. 2. Buffered Online Predictions: 1. new real data arrives (r) at time stamp i, substituting previously made prediction (o); 2. define the input vector to predict time stamp i; 3. execute prediction (t) for time stamp i; 4. compute error using predicted (t) and real (r) values; 5. back-propagate the error one single time; 6. define input vector to predict time stamp i plus one hour; 7. execute prediction of next hour (p); 8. discard oldest real data (d).

5 Experimental Evaluation

The system's modularity allows the predictive models to be, not only dynamic and easily modified, but also heterogeneous over all clusters. However, for a first evaluation of the system, we will consider homogeneous models for the prediction of the next hour electricity load.

5.1 Predictive Capacity

There are two priority concepts the system should respond. On one side, considering the expected high correlation between series of the same cluster, the system should be able to fit a neural network that represents the cluster, using the centroid of the series included in the corresponding group. On the other side, we should expect that incremental training of neural networks should produce better results when compared with models trained with historical data and no adaptation to current data. The following experiments try to answer this two questions, resulting as expected. The quality measure usually considered in electricity load forecast is the MAPE *(Mean Absolute Percentage Error)* defined as

$$MAPE = \sum_{i=1}^{n} \frac{|(\hat{y}_i - y_i)/y_i|}{n} \qquad (1)$$

where y_i is the real value of variable y at time i and $\hat{y}_i$ is the corresponding predicted value. Preliminary results on electrical power demand current data supported some of the motivations for our work. Fitted models resulted in predictions with MAPE evaluation values under 10%. We should stress that neither the clustering system's definitions nor the learning algorithm parameters were exhaustively fit. Nevertheless, Figure 3 presents an example of predictions made by a neural network for one example stream included in a cluster with hundreds of different streams, with intra-cluster correlation < 0.2, capturing the ability to train the network with the centroid of the cluster. Unfortunately, not all clusters' centroids are the best to predict the corresponding streams.

We are also interested on inspecting the advantages of online learning, comparing online training results with the predictions made by batch models. In Figure 4 we can compare the evolution of

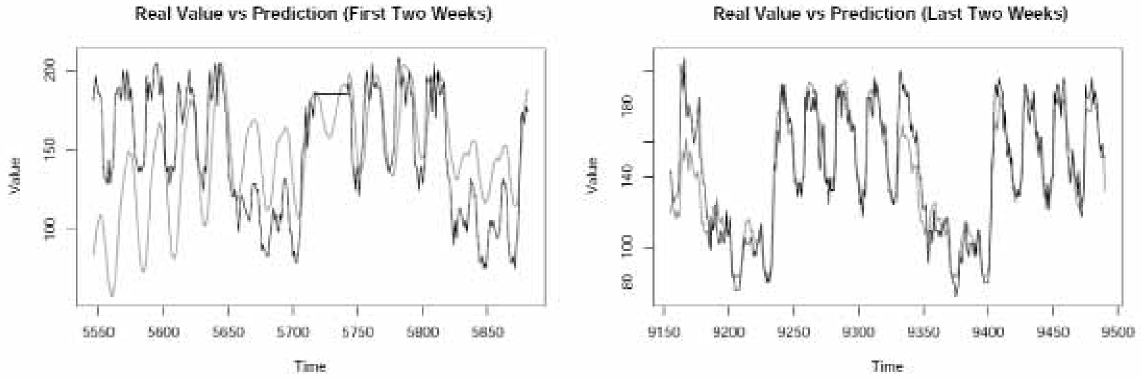

Fig. 3. Predictions for one stream belonging to a cluster with large dimension. Pictures show the plots of real values and predictions for the first two weeks after training and the last two weeks. We can see that, although the first training didn't model the stream as expected due to cluster's dimension, the online training continuously reduces the error, fitting the model to the stream, as seen in the last two weeks.

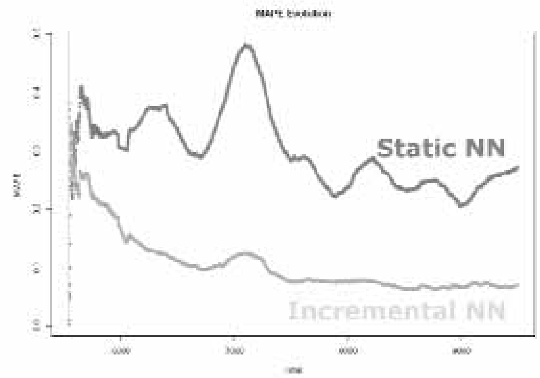

Fig. 4. MAPE evolution for the stream. The improvement achieved by incremental learning is clear, maintaining an error descent through time.

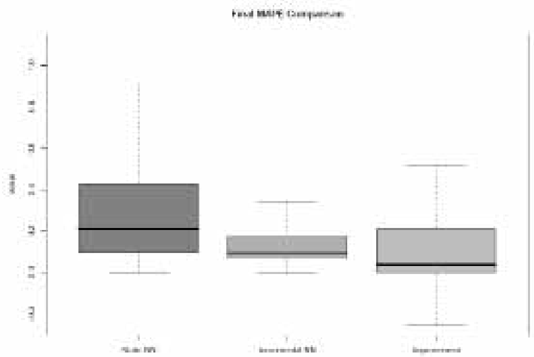

Fig. 5. Final MAPE comparison. Incremental learning of neural networks for non-null variables resulted in improvements of around 5% MAPE.

the MAPE error in the following weeks, comparing predictions made by static neural networks and incremental neural networks. The graph shows the consistency and advantages of using incremental learning. On the last week, the overall improvement achieved by online training was around 5% and is sketched on Figure 5 for all non-null variables, where positive improvement represents a decrease on the MAPE.

From this results, not only the ability to learn a model with the centroid of the group is confirmed, but also the continuously applied incremental learning is shown to favor no only the model fitting but, more important, the model adaptation to dynamic behaviors.

5.2 Comparison with another predictive strategy

To assess the quality of prediction, comparing with another predictive system, we have conducted experiments where, for a given year, the quality of the system in each month is compared with Wavelets on two precise variables, chosen as relevant predictable streams (by an expert) but expressing either low or high error. These variables are shown on Figures 6 and 7. Moreover, the

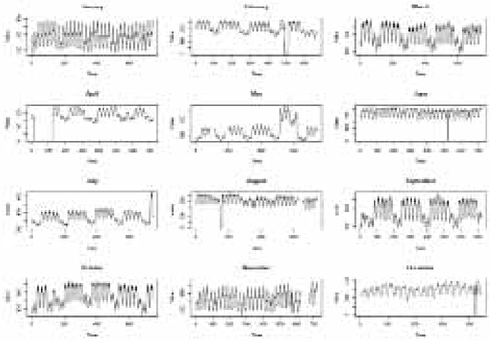

Fig. 6. Selected variables each month (low err).

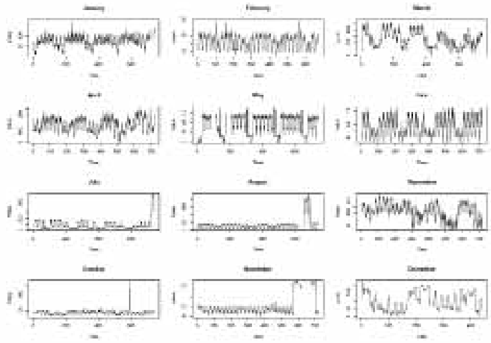

Fig. 7. Selected variables each month (high err).

	Wavelets	NNets	NNets-Wav	
	%	%	%	*p-value*
Month				
January	1.69	2.72	**1.03**	**<0.001**
February	2.99	2.79	-0.20	0.196
March	3.63	2.75	**-0.88**	**<0.001**
April	2.05	2.58	**0.53**	**0.002**
May	2.69	2.28	**-0.41**	**<0.001**
June	2.33	2.52	0.29	0.051
July	2.14	2.12	**-0.02**	**0.049**
August	2.59	2.54	-0.05	0.537
September	2.65	2.64	-0.01	0.374
October	2.28	2.36	0.08	0.127
November	2.41	2.14	-0.27	0.085
December	3.56	2.97	**-0.59**	**0.029**

Table 1. MEDAPE for selected variables (low err).

	Wavelets	NNets	NNets-Wav	
	%	%	%	*p-value*
Month				
January	9.04	10.34	**1.30**	**<0.001**
February	8.51	9.82	**1.31**	**0.002**
March	11.52	11.28	-0.24	0.166
April	9.36	12.74	**1.38**	**<0.001**
May	12.89	10.54	**-2.35**	**0.035**
June	6.68	8.10	**1.42**	**<0.001**
July	14.52	10.68	**-3.84**	**<0.001**
August	11.11	12.27	**1.16**	**0.034**
September	10.52	9.81	-0.71	0.656
October	12.45	11.25	**-1.20**	**0.002**
November	8.85	7.71	-1.14	0.356
December	11.76	10.91	**-0.85**	**0.040**

Table 2. MEDAPE for selected variables (high err).

quality measure was changed to MEDAPE *(Median Absolute Corresponding Error)*, the corresponding *median* of the APE measure, to reduce sensibility to extreme values [1]. Results are shown on Table 1, for low error variables, and Table 2, for high error variables. For the difference of the medians, the Wilcoxon [3] test was applied, and the corresponding *p-value* is shown (difference is statistically significant if $p < 0.05$). The relevance of the incremental system using neural networks is exposed, with lower error values on the majority of the studied variables. Moreover, it was noticed an improvement on the performance of the system, compared to the predictions made using Wavelets, after failures or abnormal behavior in the streams.

6 Challenging Problems

The sensor networks created by this setting can easily include thousands of sensors, each one corresponding to a given sub-network and, therefore, being object of predictive analysis. Moreover, given the evolution of hardware components, these sensors act now as fast data generators, producing information in a streaming environment. The data flow usually generated by each sensor can be extremely fast and possibly infinite in size, what corresponds to a known scenario of data stream analysis.

This setting reveals a large sensor network, where thousands of streams produce data continuously over time, representing an interesting scenario for ubiquitous computing. This problem shares some of the requirements and objectives usually inherent to ubiquitous computing. Sensors are most of the times limited in resources such as memory and computational power, and communication between them is easily narrowed due to distance and hardware limitations. Moreover, given the limited resources and fast production of data, information has to be processed in quasi-real-time, creating a scenario of multi-dimensional streaming analysis.

7 Conclusions and Future Issues

This paper introduces a system that gathers a predictive model for a large number of data variables with an horizon forecasting, incrementally constructing a hierarchy of clusters and fitting a predictive model for each leaf. The main setting of the clustering system is the monitoring of existing clusters' diameters. The main setting of the predictive strategy is the buffered online prediction of each individual variable, based on a neural network trained with clustered variables. The examples are processed as they arrive, using a single scan over the data. The system incrementally computes the dissimilarities between time series, maintaining and updating the sufficient statistics at each new example arrival. Experimental results show that the system is able to fit predictive models using the centroids of the cluster they are associated to. Moreover, applying incremental learning, using the online strategy developed in this work, seems to outperform predictions made with static predictive models.

Future work will focus on the definition of global evaluation strategy and the inspection of other online learning techniques. We believe further work on the learning task will improve the quality of neural network basic accuracy. Moreover, the electrical network spreads out geographically. The topology of the network and the position of the electrical-measuring sensors are known. From the geo-spacial information we can infer constraints in the admissible values of the electrical measures. The geo-spacial information can be used by sensors themselves. Sensors would become smart devices, although with limited computational power, that could detect and communicate with neighbors. Data mining in this context becomes ubiquitous and distributed.

Acknowledgement

Thanks to the financial support of project ALES II (POSI/EIA/55340/2004), and the FEDER Pluriannual support attributed to LIACC.

References

1. Armstrong, J. S., Collopy, F.: Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting* **8** (1992) 69–80
2. Barbará, D., Chen, P.: Using the fractal dimension to cluster datasets. In *Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining* (2000) 260–264
3. Bauer, D. F.: Constructing confidence sets using rank statistics. *Journal of American Statistical Association* **67** (1972) 687–690
4. Domingos, P., Hulten, G.: Mining high-speed data streams. In *Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining* (2000) 71–80
5. Hippert, H. S., Pedreira, C. E., Souza, R. C.: Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*, **16(1)** (2001) 44–55
6. Hoeffding, W.: Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* **58(301)** (1963) 13–30
7. Igel, C., Hüsken, M.: Improving the Rprop learning algorithm. In *Proceedings of the Second International ICSC Symposium on Neural Computation* (2000) 115–121
8. Rodrigues, P. P., Gama, J., Pedroso, J. P.: ODAC: Hierarchical clustering of time series data streams. In *Proceedings of the Sixth SIAM International Conference on Data Mining* (2006) 499–503