Electricity Load Forecast using Data Streams Techniques

Pedro Pereira Rodrigues
João Gama

LIACC-NIAAD University of Porto, Portugal

March 2007

RETINAE REal TIme Network Analysis and Enhancement (PRIME/IDEIA/70/00078)

ALES II Adaptive LEarning Systems II (POSC/EIA/55340/2004)

Load Forecast in Power Supply Networks

Load forecast is a relevant auxiliary tool for operational management of an electricity distribution network enabling:

- identification of profiles.
- prediction of picks on the demand.
- identification of **critical points** in load evolution
- necessary corrections **within available time**
- **support** for previously planned interventions
- checking the viability of **charge transfer** scenarios

Load Forecast in SCADA/DMS

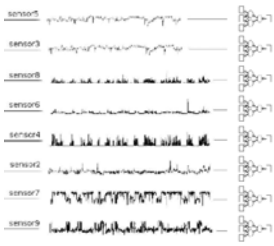
In **SCADA/DMS** (Supervisory Control and Data Acquisition / Distribution Management Systems), the load forecast functionality has to estimate, on a hourly basis, and for a **near future**, certain types of measures which are representative of system's load:

Sensor network (2500 sensors). Each sensor measures:

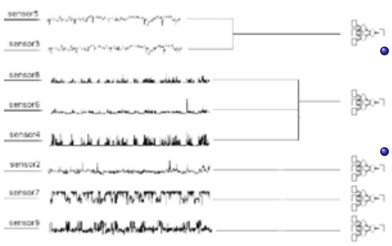
- active power
- reactive power
- current intensity

In the context of load forecast, near future is usually defined in the range of **next hours** to the limite of **seven days**.

Approach 1



Approach 2



- **similar load profiles** should be **clustered**, and
- **predictive models** should be applied to **each cluster**.

A Streaming Environment

Current SCADA/DMS systems gather a **continuous flow** of data generated at **high-speed** from sensors of an electricity distribution network.
The usual approaches for clustering and prediction use **batch procedures** which cannot cope with this streaming setting.
Our approach is in the **Data Stream** framework, maintaining in **real time** both a clustering model and a predictive model capable of:

- **incorporating** new information at the speed data arrives, and
- **detecting** changes and **adapting** the decision models to the most recent information.

Goals

Develop predictive models for groups of sensors:

- An **incremental** system is used to perform **clustering** of time series over data streams;
- At each cluster (leaf) exists an **online predictive model**;
- The system should cope with:
  - the **high-speed** and **any-time output** of the clustering structure definition and predictions;
  - the ability to **detect and adapt** to changes in the clustering structure;

With this approach, we intend to:

- **reduce or eliminate** the effort applied on configuration and training (usually slow and based on huge amounts of data)
- reach short-term predictive results with **acceptable performance**

Data Pre-Processing

- The electrical network spreads out geographically.
- The topology of the network and the position of the electrical-measuring sensors are known.
- Sensors send information at different time scales.
- Sensors act in adversary conditions: they are proneness to noise, weather conditions, battery conditions,

To reduce the impact of the noise, missing values, and different granularity, data is aggregated and synchronized in time windows of 15 minutes.

Online Divisive-Agglomerative Clustering [Rodrigues, Gama, 2006]

- Incremental system to **monitor clusters' diameters**
- Performs hierarchical clustering of **first-order differences**
- Can **detect changes** in the clustering structure
- Two Operators:
  - Expansion: expand the structure
  - Agglomeration: contract the structure
- Splitting and agglomerative criteria are supported by a confidence level given by the **Hoeffding bounds**.

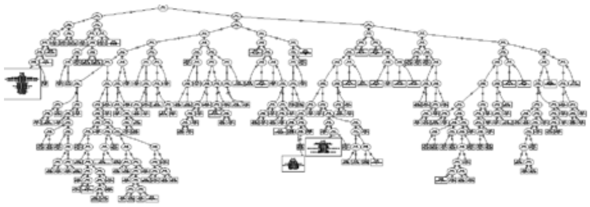
Main Algorithm [Rodrigues, Gama, 2006]

- ForEver
  - Read Next Example
  - For all the clusters
    - Update the sufficient statistics
  - Time to Time
    - Find the two farthest variables
    - Verify Merge Clusters
    - Verify Expand Cluster

Properties of ODAC

- For stationary data the cluster's diameters monotonically decrease.
- **Constant update time/memory consumption** with respect to the number of examples!
- Every time a **split** is reported
  - the **time** to process the next example **decreases**, and
  - the **space** used by the new leaves is **less than** that used by the parent.

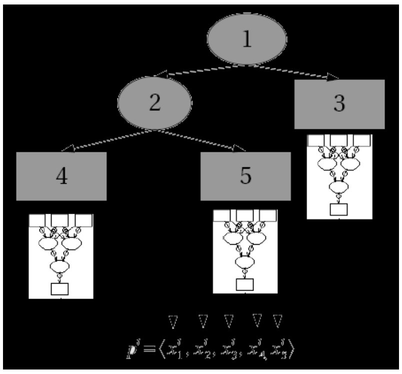
A snapshot - 1 year data, 2500 variables



Load Forecast in Data Streams

- The goal is to have an **any-time prediction** of the next-hour load value for all sensors.
- The strategy is to have **one predictive model at each cluster**, predicting all of its sensors.
- Predictive models are created for clusters which present **good behaviour**, that is, **good intracluster correlation**; for the others, the last known value is used.
- **iRprop** [Igel and Hüsken, 2000] algorithm is used to train the feedforward neural networks, with 10 inputs, 4 *tanh*-activated hidden neurons, and one linear output neuron.
- **Splitting** triggers **inheritance** of the ancestor's predictive model.
- **Aggregation** triggers **reset** of the predictive model.

Load Forecast in Data Streams

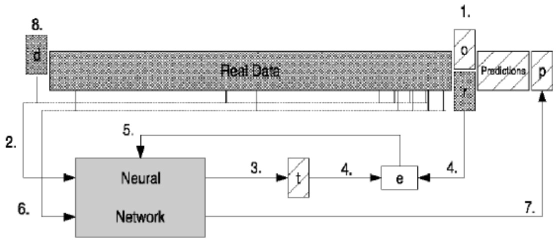


Why Neural-Nets?

- A Function approximation approach
- A 3 layer ANN can approximate any continuous function
- Fast Train and Prediction:
  - Each example is propagated once
  - The Error is back-propagated once
- No overfitting
  - First: Prediction
  - Second: Update the Model
- Smoothly adjust to gradual changes

Buffering the Input Data and the Predictions

Online prediction and training is achieved with **buffered input**.



Validation of the Clustering Approach [Rodrigues, Gama, 2006]

- For **stationary** data, ODAC performs **similar to batch** divisive analysis clustering.
- For **drifting** data, ODAC **detects and adapts** the structure to the new concepts.
- On real physiological data, ODAC resulted in the **same partitions as k-Means**.

Load Forecast of Large Sensor Networks

The system should be able to fit a **predictive model that represents the whole cluster**, training with the cluster's centroid.

Also, it is expectable that **online learning** should produce better adaptation to new examples, comparing to predictive models trained with past examples and no adaptation to current data.

Evaluation is made using the **MAPE** error measure:

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

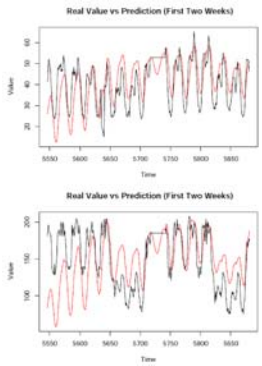
Clusters as Representatives

We have conducted experiments for current intensity measured on **over 2500 sensors** and built the clustering structure with **one year** of real data.

For each cluster which possess good intra-cluster correlation, **the system learns a predictive model using the centroid** of the corresponding time series, for a **recent period** of past data and tests them in the following weeks, for **each variable individually**.

Fitted models resulted in predictions with **MAPE evaluation values under 10%**.

Clusters as Representatives

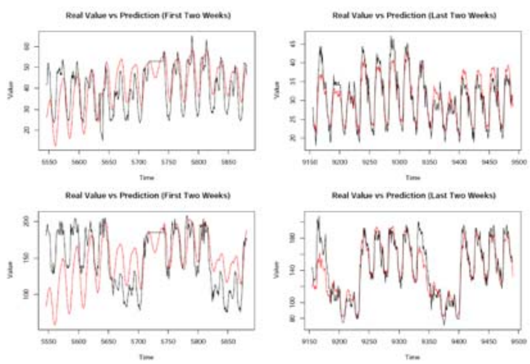


Predictions for sensor variables included in a large cluster.

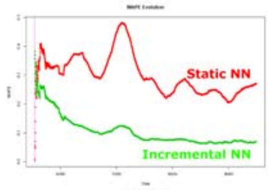
Plots present the first two weeks after initial training.

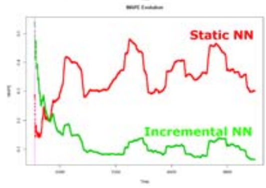
Due to the batch training, **the network is struggling to fit all the series**.

Online Learning



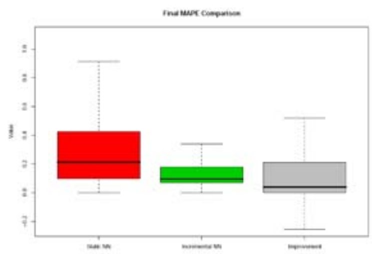
Online Learning





After some time, the incremental learning combines its efficiency with accuracy, diminishing the error **below the static model**.

Online Learning



Considering all non-null variables, after the first week, **the average improvement achieved by online training is about 5%**.

Comparison with Wavelets

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 exhibiting either **low** or **high** error.

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.

Outline Starting Setting System Description Experiments Conclusions and Open Issues

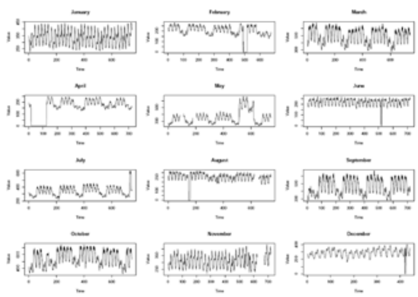


Figure: Selected variables each month (low err).

	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: MEDAPE for selected variables (low err).

Values for the median of the APE, for each sensor in their corresponding month, for predictions using **Wavelets** and our approach **(RETINAE)**.

Difference is presented between Wavelets approach and our approach, with *p-value* from Wilcoxon test.

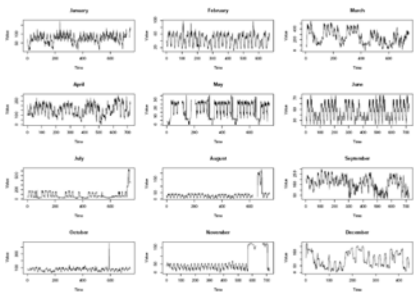


Figure: Selected variables each month (high err).

.

	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: MEDAPE for selected variables (high err).

Values for the median of the APE, for each sensor in their corresponding month, for predictions using **Wavelets** and our approach **(RETINAE)**.

Difference is presented between Wavelets approach and our approach, with *p-value* from Wilcoxon test.

Conclusions

System gathers a predictive model for a **large number** of data variables:

- Incrementally constructs a **hierarchy of clusters** fitting **one predictive model for each leaf**.
- The system has the ability to **cope with high speed** production of examples.
- The rate of predictions can be as fast as the rate of incoming examples, considering the usual rates higher than one minute.
- The system is **capable of dealing with changes** in the clustering structure.

Discussion

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**.

Comparing our system's predictions with other predictive strategies indicates **competitive performance** for the problem of load forecast.

Open Issues

Current work is concentrated on:

- improve predictions by **combining** them with the **last known value** (e.g. Kalman Filters);
- compare with **other predictive strategies** (e.g. Wavelets);
- **application** to all **three dimensions** of the electricity load;
- **testing** the system on the **24-hour forecast** and **1-week forecast** problems;
- treating **missing data** and **special events**;

Thanks for your attention!

More information:

ODAC P. P. Rodrigues, J. Gama and J. P. Pedroso. *ODAC: Hierarchical Clustering of Time Series Data Streams.* In *Proceedings of the Sixth SIAM International Conference on Data Mining*, pages 499-503. Bethesda, Maryland, USA. April 2006.

OnlineNN P. P. Rodrigues and J. Gama. *Online Prediction of Clustered Streams.* In *Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams*, pages 23-32. ECML/PKDD, Berlin, Germany. September 2006.

iRprop Christian Igel and Michael Hüsken. *Improving the Rprop learning algorithm.* In *Proceedings of the Second International ICSC Symposium on Neural Computation*, pages 115–121, Berlin, 2000.