Is Unsupervised Ensemble Learning Useful for Aggregated or Clustered Load Forecasting?

Peter Laurinec, and Mária Lucká

22.9.2017

Slovak University of Technology in Bratislava

Motivation n.1

More accurate forecast of electricity consumption is needed due to:

- Optimization of electricity consumption.
- Production of electricity. Overvoltage in grid.
- Distribution (utility) companies. Deregulation of the market. Purchase and sale of electricity.

Forecasting methods

Factors influencing electricity load:

- Seasonality (daily, weekly, ...)
- Weather (temperature, humidity, ...)
- Holidays
- Random effects

What we can use for it:

- Time series analysis methods (ARIMA, ES)
- Linear regression methods (MLR, RLM, GAM)
- AI regression methods (Trees, Neural Networks, SVR)
- Time series data mining + clustering - Creating more predictable groups of consumers[1].

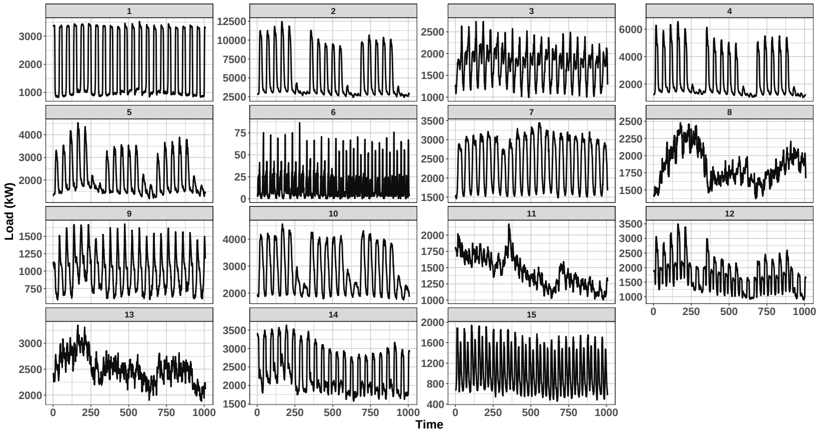
[1] Laurinec et al., ICMLDA and ICDMW, 2016

Motivation n.2 and n.3

Lot of methods are possible to use. Which one to choose?
Solution: ensemble learning.

Dynamic clustering (actually offline batch clustering) - Aggregated data from electricity consumers (clusters) changes in each sliding window.

Clusters



Solution to the problem and a target

- Clustering (generating lot of time series) + ensemble learning $\rightarrow$ classical weighting of models (based on test set) is not possible and could be slow (lot of clusters, etc.).
- Weighting of models based on training (or validation) set is possible, but it would be overfitted.
- Unsupervised ensemble learning can be promising approach.

Target and contribution: Evaluate various unsupervised ensemble learning methods combined with clustered and also simply aggregated electricity load.

Base forecasting methods

Two most important factors (assumptions) that have to be satisfied for used forecasting methods:

- Very fast to compute (i.e. week learners)
- Parametrically adaptable for various time series created by clustering of consumers.

These assumptions satisfies methods:

- CART (RPART) tree
- CTREE - Conditional inference trees
- ARIMA
- Exponential smoothing

Detailed description of forecasting methods

RPART

- Simple daily and weekly attributes

CTREE.lag

- Lag feature (denoised); daily and weekly attributes in sinus and cosine form.

CTREE.dft

- Daily and weekly attributes in form of Fourier transformation (6 and 12 pairs).

STL+ARIMA, STL+ES, ES

- STL decomposition in combination with ARIMA and ES; ES standalone.

Unsupervised ensemble learning

The 1st possibility is to do ensemble learning on just base forecasting methods.

Bootstrapping time series:

- Classical sampling from training set (bagging) - for regression (RPART, CTREE)
- Moving block bootstrap (mbb) - for time series analysis methods (ARIMA, ES)

Evaluated was the median from each of a bootstrapped method.

Bootstrapping

For regression tree methods:

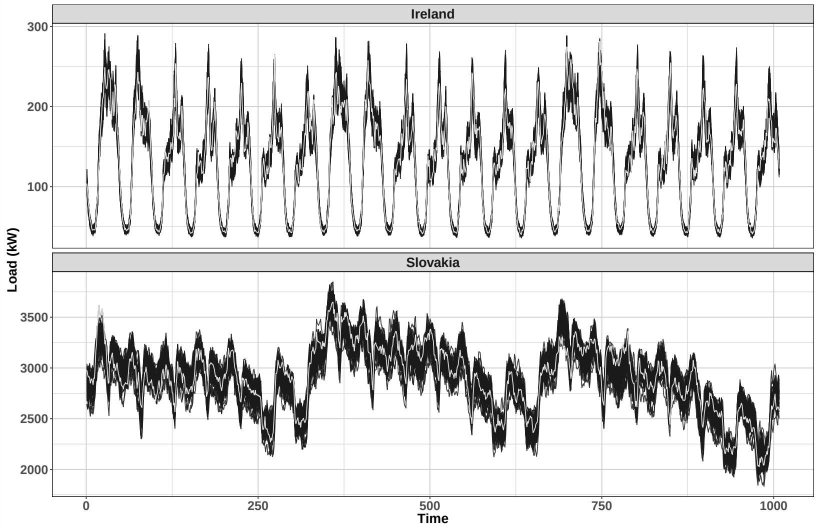
- The sample ratio was randomly sampled in the range of 0.7 – 0.9.
- Sampled was also maximal depth, complexity parameter, minimal criterion

For time series analysis methods:

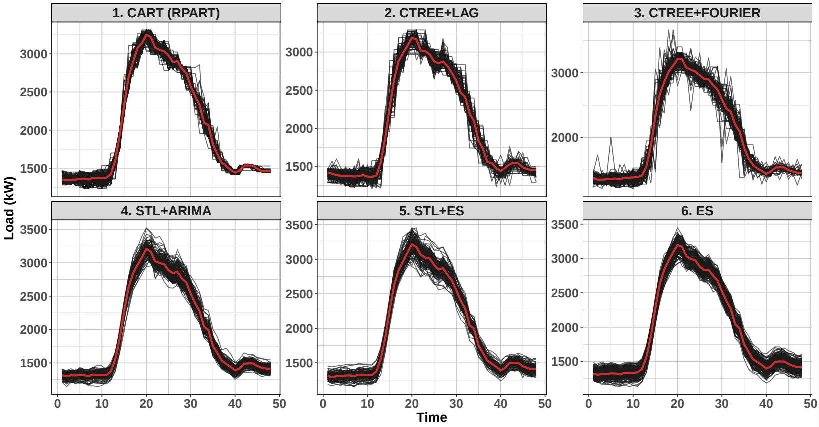
- Box-Cox transformation, STL decomposition, remainder is bootstrapped by mbb, inverse Box-Cox [2]

[2] Bergmeir, Hyndman, Benítez, Int. J. of Forecasting, 2016

Moving block bootstrapping



Results of Bagging



Unsupervised ensemble learning

The 2nd possibility is to combine all bootstrapped forecasts from all methods.

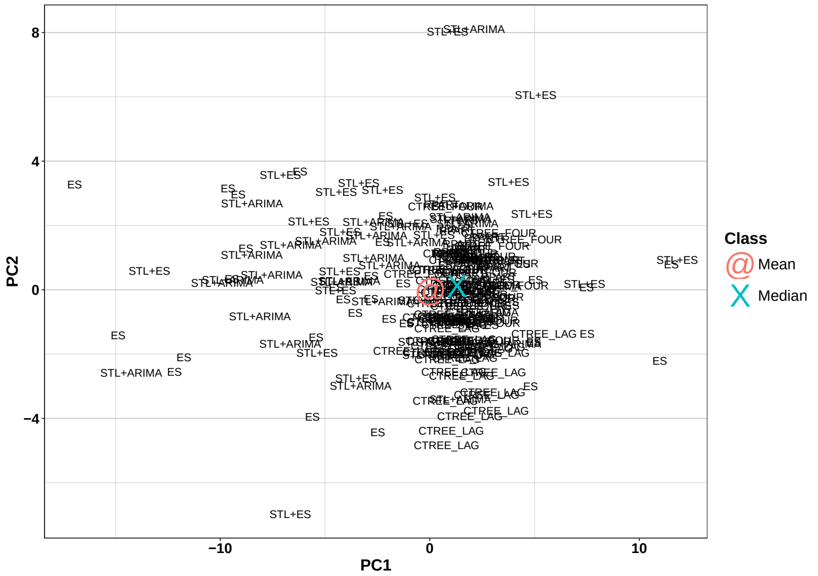
Unsupervised (structure-based) possibilities for aggregating bootstrapped forecasts:

- Simple mean or median
- Averaging by methods
- Clustering

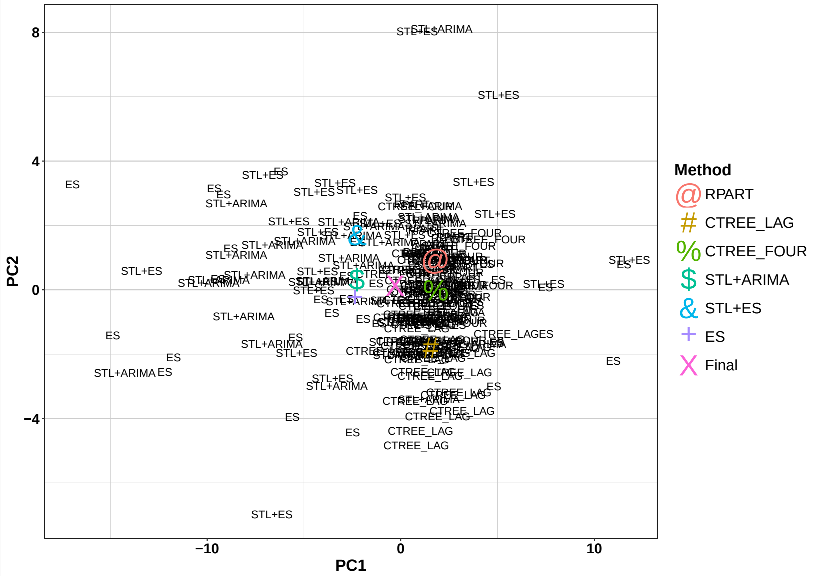
Aggregating from all created bootstrappings:

a) Simple aggregation based - average and median
b) Naive cluster based - average of medians of methods
c) Cluster-based - K-means-based, DBSCAN-based and OPTICS-based

Simple aggregation based methods



Naive cluster based method



Cluster-based approaches

PCA is used to extract three principal components (from 48 dim.) in order to reduce noise.

K-means-based:

- K-means clustering with K-means++ centroid initialization.
- Optimal number of clusters in the range of $3 - 8$ by DB index.
- Centroids were averaged to the final ensemble forecast.

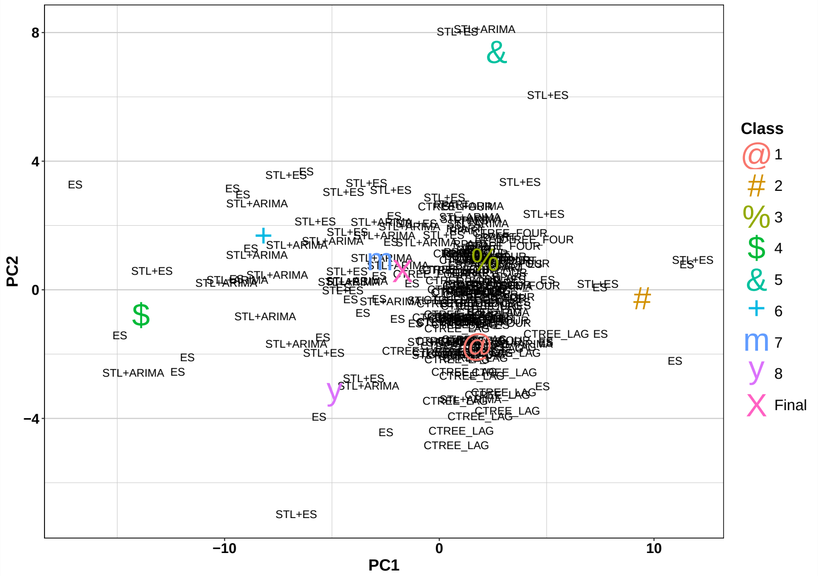
DBSCAN-based:

- requires two parameters: ϵ (set to 1.45) and *minPts* (set to 8).
- Ensemble forecast is created by the average of medians of clusters.

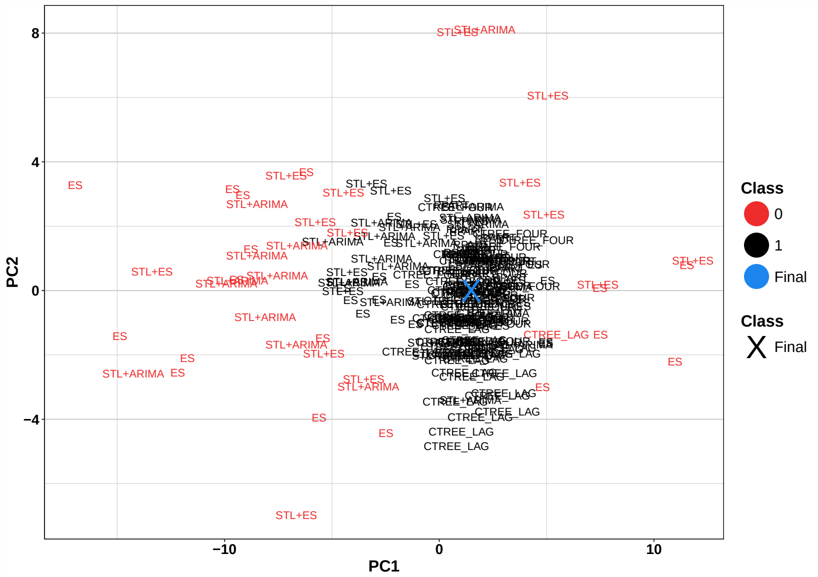
OPTICS-based:

- Automatic ξ-cluster procedure. ξ defines the degree of steepness (set to 0.045), which is applied in the so-called reachability plot of distances.
- The final ensemble forecast is the median of medians of clusters.

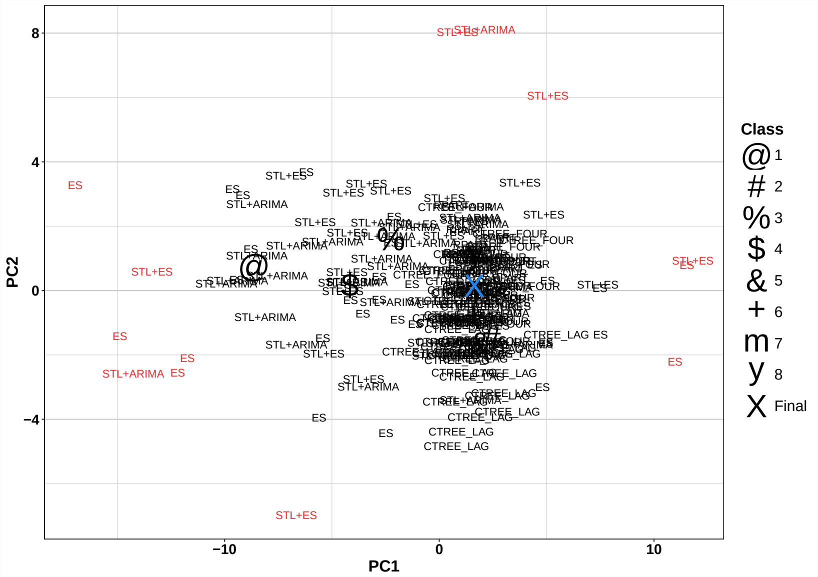
K-means-based method



DBSCAN-based method



OPTICS-based method



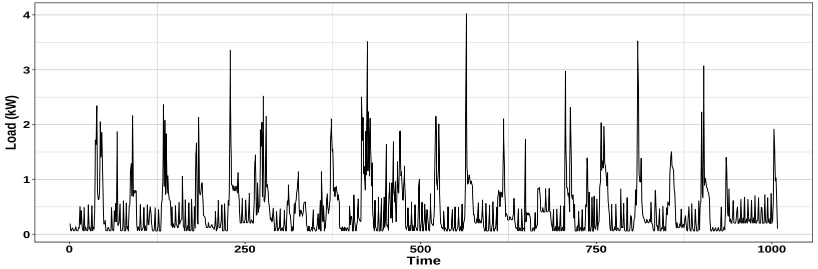
Aggregation with clustering

1. Set of time series of electricity consumption of the length of two weeks
2. Normalization (z-score)
3. Computation of representations of time series by MLR (extraction of D. and W. reg. coeff.)
4. Automatic determination of optimal number of clusters K (DB-index)
5. Clustering of representations (K-means with centroids initialization by K-means++)
6. Summation of K time series by found clusters
7. Training of K forecast models and the following forecast
8. Summation of forecasts and evaluation
9. Remove first day and add new one to the training window (sliding window approach), go to step 1

Data from smart meters

Ireland residences

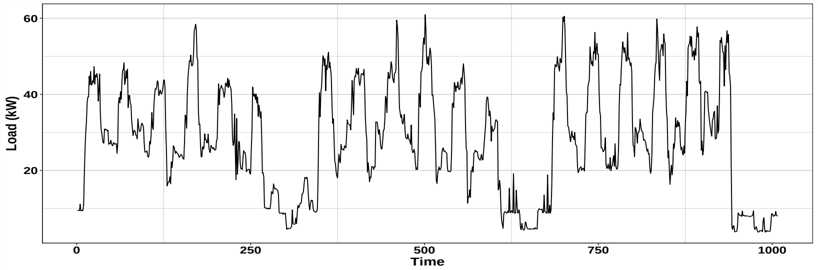
- 3639 consumers. Residences.
- 48 measurements per day.
- Test set from three months of year 2010 (February, May and September).



Data from smart meters

Slovak factories

- 3630 consumers. Factories and enterprises.
- 96 measurements per day. Aggregated to 48.
- Test set from three months of years 2013 and 2014 (September, February and March, June).



Evaluation of forecast

The accuracy of the forecast of electricity consumption was measured by MAPE (Mean Absolute Percentage Error).

$$\text{MAPE} = 100 \times \frac{1}{n} \sum_{t=1}^{n} \frac{|x_t - \overline{x}_t|}{x_t},$$

where x_t is a real consumption, $\overline{x}_t$ is a forecasted load and n is a length of the time series.

Results - ensembles

	Agg.-Irel.	Clust.-Irel.	Agg.-Slov.	Clust.-Slov.
CART.bagg	3.7908	3.7964	3.1561	3.0993
CTREE.bagg.lag	3.8081	3.7599	**2.9568**	**2.8730**
CTREE.bagg.dft	**3.6746**	**3.7103**	3.0080	2.9341
STL+ARIMA.mbb	3.9344	3.9085	3.0325	2.9993
STL+ES.mbb	3.9901	4.0221	3.0306	3.0021
ES.mbb	4.0565	4.0723	2.9760	2.9446
Average	3.7034	3.6970	2.8312	2.8086
Median	3.6103	3.6046	2.8329	2.7980
AveMedians	3.6704	3.6771	2.8179	2.7901
K-means	4.3018	4.0189	2.9715	3.0916
DBSCAN	3.9752	3.7985	2.9352	2.7532
OPTICS	3.7482	3.7239	2.9253	2.7982

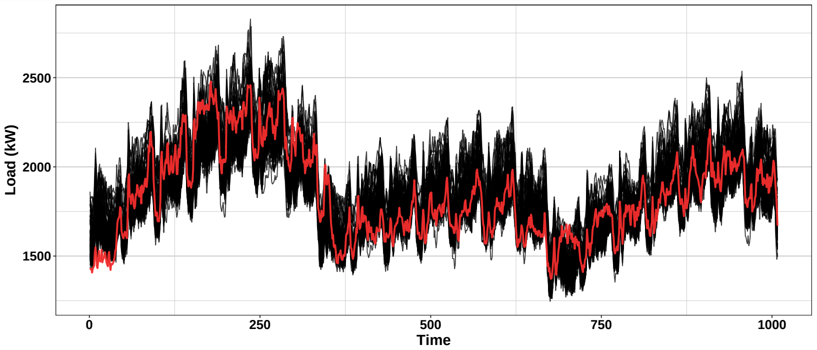
Agg.-Irel.	Clust.-Irel.	Agg.-Slov.	Clust.-Slov.
0.0011	$<$0.0001	0.1379	0.2415

Results - base vs. ensemble

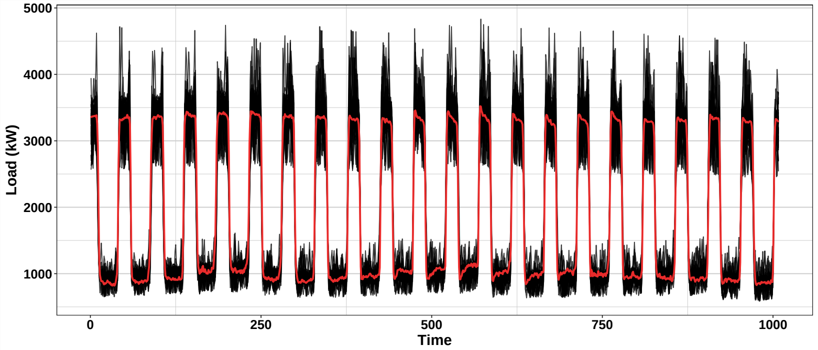
	Agg.-Irel.	Clust.-Irel.	Agg.-Slov.	Clust.-Slov.
CART	**3.8570**	3.8502	3.1921	3.1416
CTREE.lag	3.9203	**3.7523**	2.9950	2.8954
CTREE.dft	4.0849	3.9214	3.1944	3.0096
STL+ARIMA	4.0718	3.8943	2.7567	2.7404
STL+ES	4.2750	4.1866	2.6887	2.6424
ES	4.8000	4.2219	2.3957	**2.4672**

	Agg.-Irel.	Clust.-Irel.	Agg.-Slov.	Clust.-Slov.
CART.bagg	0.1113	0.0760	**0.0075**	**0.0002**
CTREE.bagg.lag	**0.0001**	0.4807	**0.0004**	**0.0036**
CTREE.bagg.dft	$<$**0.0001**	$<$**0.0001**	$<$**0.0001**	**0.0002**
STL+ARIMA.mbb	**0.0178**	0.0592	0.9875	0.9996
STL+ES.mbb	**0.0380**	**0.0330**	0.9980	0.9995
ES.mbb	$<$**0.0001**	0.2559	1.0000	1.0000

When time series bootstrapping fails



When time series bootstrapping fails



Conclusion

- The **simple median aggregation** of bootstrapped forecasts is very good approach.
- **Clustering-based ensembles** is not always the best approach.
- The **bagging** using **STL** decomposition and **Box-Cox** transformation **fails** when data are noisy.
- **Exponential smoothing state space** models - follow it, use it! Can be combined with the Temporal Hierarchical Forecasting as well.

Future work:

- Develop **more robust bagging** methods for **time series**.
- Develop **more automatic** techniques for **density-based clustering**.