STATS 207: Time Series Analysis Autumn 2020

Lecture 20: Course Recap; High-Dimensional Data, DeepAR, and Vest.

Dr. Alon Kipnis

November 20th 2020

Genera Info

- Final Assessment is Due Friday at 11:59 PM.
- Course feedback evaluation on Axes.

Course Recap

List of Topics, I

1. Introduction:
    - Examples of time series and time series models
    - Theoretical constructs:
        - Mean function
        - Autocovariance and autocorrelation functions (ACF)
        - Cross-covariance and cross-correlation function (CCF)
        - Stationarity and joint stationarity
        - Sample ACF and CCF, Confidence limits
    - Classical Regression:
        - LS solution
        - t-test for regression parameters
        - Competing models, explainable variance, F-test
        - Coefficient of determination (R^2)
        - Periodogram as explainable variance of sinusoids
    - Data wrangling, trend models, and data smoothing
        - Detrending via a trend model
        - Detrending via differencing
        - Log and power transformations
        - Smoothing (moving average, kernel, loess, smoothing splines)

List of Topics, II

2. ARMA Modeling:
    - AR and MA
    - Causality and invertibility
    - Forecasting in ARMA models
    - Estimating ARMA parameters (Yule-walker, ML, conditional LS)
    - ARIMA
    - SARIMA
    - Model diagnostics (Ljung-Box)
    - Regression with autocorrelated errors
    - Lagged Regression (using transfer function modeling)
    - Volatility models: ARCH and GARCH

List of Topics, III

3. Spectral Analysis
   - Periodogram
   - Spectral density (meaning of the term 'white noise')
   - Linear filtering and spectrum
   - Spectral estimation: smoothing the periodogram
   - Cross-spectrum and coherency
   - Frequency-domain regression (coherency, competing models)
   - Spectral principal components
4. State-space modeling
   - State-space equations
   - State estimation (Kalman filter and smoother)
   - Estimating state-space models (ML and EM)
   - Seasonal decomposition
   - State-space models with switching (stochastic volatility)
   - Bayesian analysis of state-space models

High-Dimensional Data

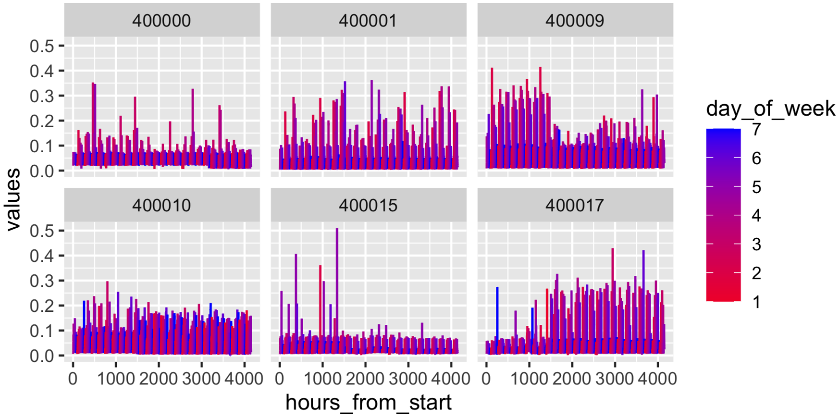
Dataset I: `traffic`

- **Occupancy rate** (between 0 and 1) of 172 different car lanes of the San Francisco bay area freeways across time.
- 15 months worth of hourly data.
- **Source**: California Department of Transportation, www.pems.dot.ca.gov.

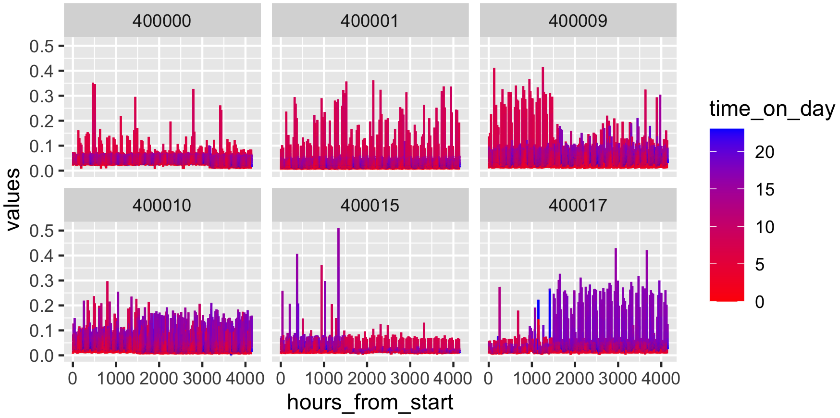
```
summary(traffic %>% select(id, values, sensor_day, time_on_day))
```

```
      id             values          sensor_day      time_on_day
 Min.   :400000   Min.   :0.00000   Min.   :  0.00   Min.   : 0.0
 1st Qu.:400485   1st Qu.:0.02190   1st Qu.: 43.00   1st Qu.: 6.0
 Median :400991   Median :0.04638   Median : 86.00   Median :12.0
 Mean   :401018   Mean   :0.05296   Mean   : 86.02   Mean   :11.5
 3rd Qu.:401580   3rd Qu.:0.07053   3rd Qu.:129.00   3rd Qu.:18.0
 Max.   :402090   Max.   :1.00000   Max.   :172.00   Max.   :23.0
 hours_from_start
 Min.   :   1
 1st Qu.:1038
 Median :2076
 Mean   :2076
 3rd Qu.:3114
 Max.   :4151
```

Example: `traffic` **by** `day_of_week`



Example: `traffic` by `time_of_day`



Dataset II: Walmart's `sales`

Daily unit sales per **product and store**:

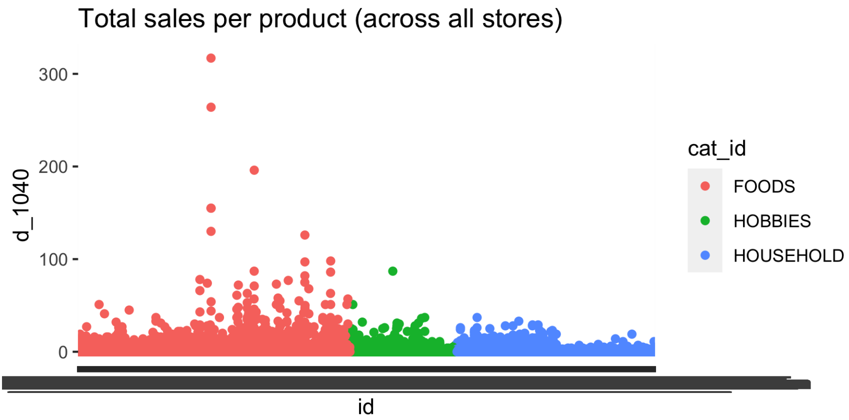
- $\boldsymbol{x}_{ti}$ is the number of items of product i sold at day t.
- $i = 1, \ldots, 30490$: 3049 different products across 10 different stores $= 30490$ unique (product_id,store_id)
- $t = 1, \ldots, 1941$ (5.3 years).
- M5 competition: require 28-days-ahead forecast (`https://www.kaggle.com/c/m5-forecasting-accuracy`)

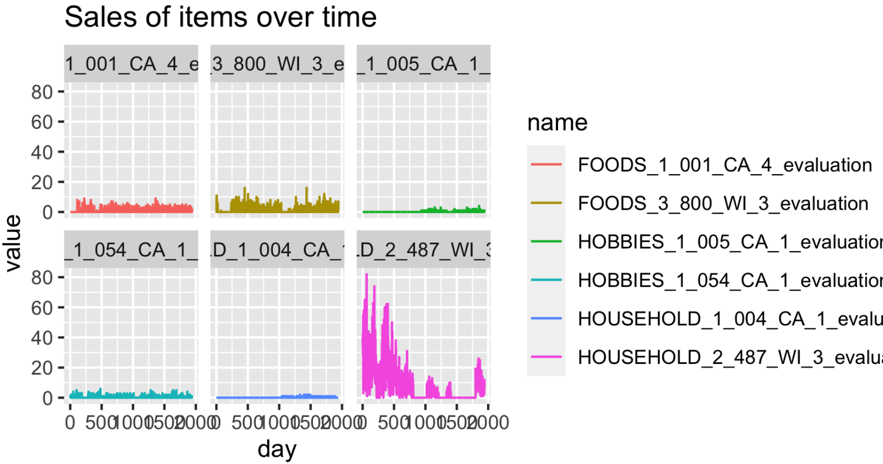
Walmart's `sales`

```r
summary(sales)
```

```
                id                             cat_id          store_id
 FOODS_1_001_CA_1_evaluation:     1   FOODS    :14370   CA_1   : 3049
 FOODS_1_001_CA_2_evaluation:     1   HOBBIES  : 5650   CA_2   : 3049
 FOODS_1_001_CA_3_evaluation:     1   HOUSEHOLD:10470   CA_3   : 3049
 FOODS_1_001_CA_4_evaluation:     1                     CA_4   : 3049
 FOODS_1_001_TX_1_evaluation:     1                     TX_1   : 3049
 FOODS_1_001_TX_2_evaluation:     1                     TX_2   : 3049
 (Other)                    :30484                      (Other):12196
 state_id       d_1                d_2                 d_3
 CA:12196   Min.   :  0.00   Min.   :  0.000   Min.   :  0.00
 TX: 9147   1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.:  0.00
 WI: 9147   Median :  0.00   Median :  0.000   Median :  0.00
            Mean   :  1.07   Mean   :  1.041   Mean   :  0.78
            3rd Qu.:  0.00   3rd Qu.:  0.000   3rd Qu.:  0.00
            Max.   :360.00   Max.   :436.000   Max.   :207.00
```

sales – Total Sales at a Given Day




Sales of items over time
1_001_CA_4_e
3_800_WI_3_e
_1_005_CA_1_
_1_054_CA_1_
D_1_004_CA_
D_2_487_WI_3
80
60
40
20
0
value
0
500
1000
1500
2000
day
name
FOODS_1_001_CA_4_evaluation
FOODS_3_800_WI_3_evaluation
HOBBIES_1_005_CA_1_evaluation
HOBBIES_1_054_CA_1_evaluation
HOUSEHOLD_1_004_CA_1_evalu
HOUSEHOLD_2_487_WI_3_evalu

Challenges

- The standard task is **prediction**:

$$\hat{\boldsymbol{y}}_{t+m}^{t} = \hat{\boldsymbol{y}}_{t+m}^{t}(\boldsymbol{y}_{1:t}).$$

- Prediction can be assisted by regression over **exogenous features series**:

$$\hat{\boldsymbol{y}}_{t+m}^{t} = \hat{\boldsymbol{y}}_{t+m}^{t}(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{1:t}).$$

  Example: Calendar data

$$\boldsymbol{x}_t = (\texttt{hour\_in\_day}_t, \texttt{day\_of\_week}_t, \texttt{month\_in\_year}_t, \texttt{holiday}_t)'.$$

- **Special properties**:
  - Dependencies between many scalar time series.
  - Sparsity.
  - Discrete variables.
- **Notable Techniques**
  - Dimensionality reduction using principal component analysis.
  - Large regression models. Holdout data for validation and testing.

DeepAR

DeepAR

DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks

David Salinas, Valentin Flunkert, Jan Gasthaus
Amazon Research
Germany
<dsalina,flunkert,gasthaus@amazon.com>

Abstract

Probabilistic forecasting, i.e. estimating the probability distribution of a time series' future given its past, is a key enabler for optimizing business processes. In retail businesses, for example, forecasting demand is crucial for having the right

DeepAR – Model

- **Data**:
  - $(\boldsymbol{z}_t)$ is a vector time series, known for $j = 1, \dots, t$
  - $(\boldsymbol{x}_t)$ is a vector time series of **covariates**, known for all $t = 1, \dots, T$.
- **Goal:** Model **one-step-ahead posterior** (or more) given the past

$$\Pr(z_{t+1} | \boldsymbol{z}_{1:t}, \boldsymbol{x}_{1:T})$$

- **Method**: Suppose

$$\Pr(z_{t+1} | \boldsymbol{z}_{1:t}, \boldsymbol{x}_{1:T}) = \ell\left(z_{t+1} | \theta(\boldsymbol{h}_{t+1})\right)$$

where

- ℓ is a likelihood function with parameters θ.
- $\theta_{i,t} \equiv \theta(\boldsymbol{h}_{i,t})$ depends on the output of a **recurrent neural network**

$$\boldsymbol{h}_t = h_\Theta(\boldsymbol{h}_{t-1}, z_{t-1}, \boldsymbol{x}_t)$$

(h_Θ is a multi-layer recurrent neural network with LSTM cells)

Likelihood Function – Examples

- Example: Gaussian likelihood

$$\ell_G(z|\mu,\sigma) = (2\pi\sigma^2)^{-1/2} e^{-\frac{(z-\mu)^2}{2\sigma^2}},$$

$$\mu = \mu(\boldsymbol{h}_t) = \boldsymbol{w}'_\mu \boldsymbol{h}_t + b_\mu, \qquad \sigma^2 = \sigma^2(\boldsymbol{h}_t) = f(\boldsymbol{w}'_\sigma \boldsymbol{h}_t + b_\sigma).$$

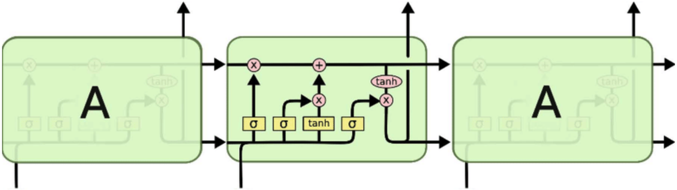
- Example: Negative binomial likelihood

$$\ell_{NB}(z|\mu,\alpha) = \frac{\Gamma(z+\frac{1}{\alpha})}{\Gamma(z+1)\Gamma(\frac{1}{\alpha})} \left(\frac{1}{1+\alpha\mu}\right)^{\frac{1}{\alpha}} \left(\frac{\alpha\mu}{1+\alpha\mu}\right)^{z},$$

$$\mu = \mu(\boldsymbol{h}_t) = f(\boldsymbol{w}'_\mu \boldsymbol{h}_t + b_\mu), \qquad \alpha = \alpha(\boldsymbol{h}_t) = f(\boldsymbol{w}'_\alpha \boldsymbol{h}_t + b_\alpha).$$

Recurrent Neural Networks

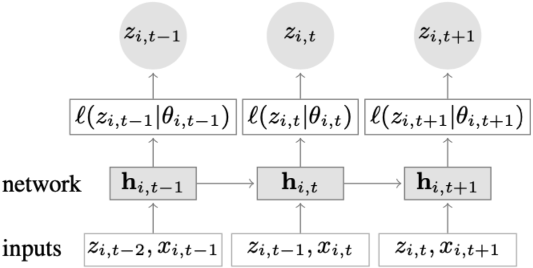
- Multilayer recurrent neural network with LSTM cells:



Model Estimation (training)

- **Assume**: We have **many** past+future data instances:

$$(\boldsymbol{z}_{i,1:t+1})_{i=1,\ldots,N}, \qquad (\boldsymbol{x}_{i,1:T})_{i=1,\ldots,N}.$$



- Log-likelihood

$$L(\Theta) = \sum_{i=1}^{N} \log \ell\left(z_{i,t+1} | \theta(\boldsymbol{h}_{i,t'+1}; \Theta)\right)$$

- Minimize $L(\Theta)$ using stochastic **gradient descent**.

How to generate many training instances?

- Take many length-$(t'+1)$ windows from available data:

$$\boldsymbol{z}_{i,1:p+1} = \{z_{i+1}, \dots, z_{i+t'}\}, \quad i = 1, \dots, t - t'.$$

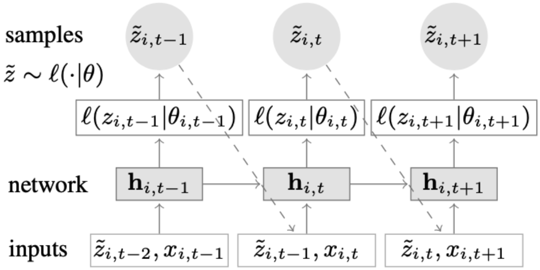
- Log-Likelihood:

$$L(\Theta) = \sum_{i=1}^{N} \log \ell \left(z_{i,t'+1} | \theta(\boldsymbol{h}_{i,t'+1}; \Theta) \right)$$

  Note: Network weights Θ do not depend on time (likelihood parameters θ do depend on time).

- Implicit **stationarity** assumptions!

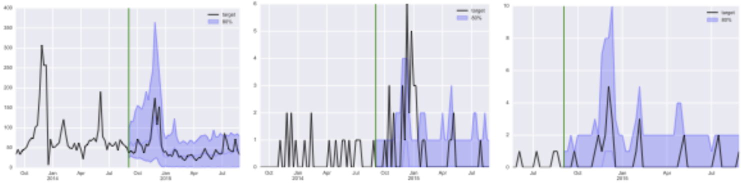
Prediction



samples
$\tilde{z} \sim \ell(\cdot|\theta)$
$\tilde{z}_{i,t-1}$
$\tilde{z}_{i,t}$
$\tilde{z}_{i,t+1}$
$\ell(z_{i,t-1}|\theta_{i,t-1})$
$\ell(z_{i,t}|\theta_{i,t})$
$\ell(z_{i,t+1}|\theta_{i,t+1})$
network
$\mathbf{h}_{i,t-1}$
$\mathbf{h}_{i,t}$
$\mathbf{h}_{i,t+1}$
inputs
$\tilde{z}_{i,t-2}, x_{i,t-1}$
$\tilde{z}_{i,t-1}, x_{i,t}$
$\tilde{z}_{i,t}, x_{i,t+1}$

DeepAR – Examples

- Sales prediction:



DeepAR as a non-linear State-Space Model

- **Observation Equation**:

$$z_t = \mu(\boldsymbol{h}_t) + v_t, \qquad R = \sigma^2(\boldsymbol{h}_t).$$

  (linear in $\boldsymbol{h}_t$; state-dependent heteroscedasticity)

- **State Dynamics**:

$$\boldsymbol{h}_t = h_\Theta\left(\boldsymbol{h}_{t-1}, z_{t-1}, \boldsymbol{x}_t\right).$$

  (non-linear state-dynamics)

- In essence, the **main innovation** is a **multilayer LSTM** modeling of **state-dynamics**.
- **Called-upon comparison**: DeepAR vs. a linear state-space model. Can **multilayer LSTM** improve over a **linear model**?

DeepAR as a non-linear State-Space Model (cont'd)

Advantages (according to the authors):

> tages compared to classical approaches and other global methods: (i) As the model learns seasonal behavior and dependencies on given covariates across time series, minimal manual feature engineering is needed to capture complex, group-dependent behavior; (ii) DeepAR makes probabilistic forecasts in the form of Monte Carlo samples that can be used to compute consistent quantile estimates for all sub-ranges in the prediction horizon; (iii) By learning from similar items, our method is able to provide forecasts for items with little or no history at all, a case where traditional single-item forecasting methods fail; (vi) Our approach does not assume Gaussian noise, but can incorporate a wide range of likelihood functions, allowing the user to choose one that is appropriate for the statistical properties of the data.
>
> ing the scale-free nature (approximately straight line) present in the `ec` dataset (axis labels omitted due to the non-public nature of the data).

In fact:

(I) Main benefit: Model automatically learns complicated connections.

(II) Not a **distinguishing feature** of DeepAR.

(III) A feature of the state-space/vector formulation.

(IV) Not unique to DeepAR (Generalized linear model formulation).

VEST

VEST

arXiv.org > stat > arXiv:2010.07137

Search...

Help | Adv

Statistics > Machine Learning

[Submitted on 14 Oct 2020]

VEST: Automatic Feature Engineering for Forecasting

Vitor Cerqueira, Nuno Moniz, Carlos Soares

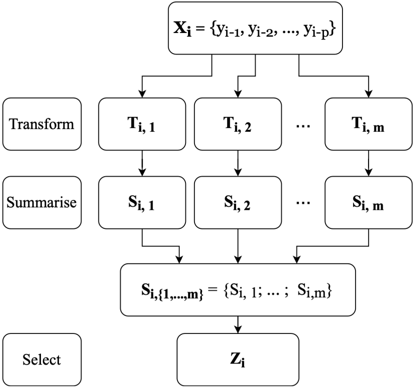
Time series forecasting is a challenging task with applications in a wide range of domains. Auto-regression is one of the most common approaches to address these problems. Accordingly, observations are modelled by multiple regression using their past lags as predictor variables. We investigate the extension of auto-regressive processes using statistics which summarise the recent past dynamics of time series. The result of our research is a novel framework called VEST, designed to perform feature engineering using univariate and numeric time series automatically. The proposed approach works in three main steps. First, recent observations are mapped onto different representations. Second, each representation is summarised by statistical functions. Finally, a filter is applied for feature selection. We discovered that combining the features generated by VEST with auto-regression significantly improves forecasting performance. We provide evidence using 90 time series with high sampling frequency. VEST is publicly available online.

VEST (Vector Statistics from Time series)

- Map **recent observations** to **many different features**.
- Use these features for prediction.
- Develop a procedure as **automatic** as possible.

- Advantage of feature-based inference over end-to-end approaches: **interpretability**.

VEST – Feature Engineering Workflow



- $T_{i,j} \in \mathbb{R}^q$ is the j-th **transformation** of $\boldsymbol{X}_i$ (e.g., `diff`)
- $S_{i,j} \in \mathbb{R}^{q_j}$ is the j-th **summary** of $\boldsymbol{X}_i$ (e.g., `mean`, `max`, $\overline{\texttt{ACF}}$)

VEST – Transformations

Table 1: Transform operations used in VEST.

Operation	Description
I	The Identity transformation, in which each X is mapped onto itself
SMA	We apply a one-sided simple moving average which can be beneficial to smooth out spurious fluctuations and highlight the general trend. The number of periods is equal to the square root of the length of X, rounded to the nearest unit
DIFF	First differences are applied to transform the original embedding vector into one without trend. This transformation can help with the modelling of time series with a strong trend component
DIFF2	Second differences, which is equivalent to applying the **DIFF** operation twice to X_i. This transformation is useful for describing the curvature of the data
BC	Box-Cox transformation, for stabilising the variance of the time series. The transformation parameter is optimised using all the available observations according to Guerrero [14] (minimizing the coefficient of variation)
SIN	Sine terms of order 1 of the Fourier series. This transformation captures the seasonality of the time series. We remark that the frequency of the time series must be available to compute these terms
COS	Similar and complementary to SIN, COS captures the cosine terms of order 1 of the Fourier series.
DWT	We apply a 1-level discrete wavelet transform using the Daubechies wavelet [36], and retrieve the coefficients of the respective detail signal

VEST – Summary Operations, I

Table 2: Summary operations used in VEST.

Operation	Description
MEAN	Arithmetic mean, which is used to estimate the average level of the vector
MDN	Median: similar to the mean, but more robust to outliers
SD	Standard deviation, as a measure of the overall dispersion in the vector
VAR	Variance of the vector, which also measures dispersion
IQR	Inter-quartile range, which is another measure of dispersion of the data, but more robust to outliers
RD	Relative Dispersion, which is estimated according to the ratio between the standard deviation of the vector and the standard deviation of the differenced vector [46]
MIN	Minimum value of the vector
MAX	Maximum value of the vector
LP	Last known point of the vector
SK	Skewness of the distribution of the vector, which is a measure of its asymmetry [46]
KRT	Kurtosis for describing the flatness of the data with respect to a normal distribution [46]
P05, **P95**	The 5th and 95th percentiles of the vector
ACC_1, **ACC_2**	Average (**ACC_1**) and standard deviation (**ACC_2**) of the acceleration of the vector, estimated according to the ratio between the simple moving average and the exponential moving average of equal period. In our experiments, the period for computing the moving averages was set to the squared root of the length of the vector, rounded to units

VEST – Summary Operations, II

BP	Level of auto-correlation, which is estimated using a Box-Pierce test statistic [4,46]
PACF	Average value of the partial auto-correlation function of the vector up to 10 lags
ACF	Average value of the auto-correlation function of the vector up to 10 lags
LRD1 **LRD2**	Long-range dependence, estimated using the Hurst exponent approach with wavelet transform with 1 (**LRD1**) and 2 moments (**LRD2**) [46]
SLP	Slope of the vector which describes its overall steepness [39]
NORM	Euclidean norm of the vector, which captures its total energy
NO	Number of outliers, estimated according to the number of observations above or below 1.5 times the inter-quartile range
AMP	Average amplitude of the fast Fourier transform of the vector
STEP	Binary random variable which denotes the presence of a step change [29]. This statistic detects structural breaks in the data
PEAK_I, PEAK_D	Number of local maxima (**PEAK_I**) and local minima (**PEAK_D**) in the vector [29]. These statistics describe the level of oscillation of the data
OD	Overall direction of the vector, estimated by the difference between the number of times the vector increases and the number of times the vector decreases
PV_ST, PV_LT	Short-term and long-term variability, respectively, estimated using the Poincaré plot [5]
MLE	Maximum Lyapunov exponent, which quantifies the chaotic level of a time series [46]

VEST – Feature Selection

- Concerns:
  - Some features **do not provide useful information** for forecasting.
  - Features may be **highly correlated** with each other.
- **Simple** selection rules:
  - Remove features with a low number of unique values.
  - If two features are highly correlated, remove one of them.
- More complicated selection rules are available.

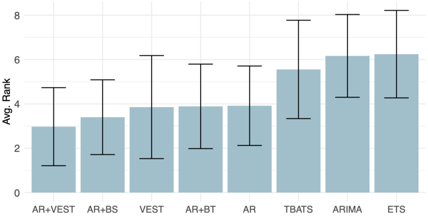
VEST – Modeling and Evaluation

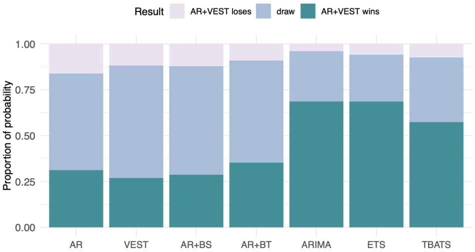
- **Data** for fitting: Many variants for combining $\{\boldsymbol{X}\}_{i=1}^{n}$ series with feature-selected series $\{\boldsymbol{Z}_i\}_{i=1}^{n}$.
- **Most successful strategy** reported in (Cerqueira et. al. 2020) is `AR+VEST`: fit a vector AR to

$$\boldsymbol{U}_i = [\boldsymbol{X}_i, \boldsymbol{Z}_i].$$

- **Performance evaluation**:
    - **90 different time series**, each with at least 1,000 observations.
    - **Holdout**: 60% training, 20% validation (for parameter optimization), 20% testing.

VEST – Results





The End!


When you finish a course
on time series analysis
The laws of time are mine
Statisticians be like