Contents

1. Time series decomposition: TSD, STL

Typical TSD (time series decomposition) looks like:

$$y_t = T_t + S_t + R_t$$

T_t – trend, S_t – seasonality component, R_t – random fluctuations, a.k.a. noise.

The decomposition can also take on the following forms:

$$y_t = T_t S_t R_t, \text{ or } \; y_t = (T_t + S_t) R_t.$$

1. 1. Classical TSD (using moving averages)

Moving average (MA) is given by the following expression:

$$\mathrm{MA}(y_t; m) = \frac{1}{m} \sum_{j=-k}^{k} y_t,$$

where $m = 2k + 1$ is called *window size* and has to be odd. Backward formula:

$$\mathrm{MA}(y_t; m) = \frac{1}{m} \sum_{j=-m}^{0} y_t,$$

Forward formula:

$$\mathrm{MA}(y_t; m) = \frac{1}{m} \sum_{j=0}^{m} y_t.$$

For $m = 4$:

$$\mathrm{MA}(y_t; 4) = \frac{1}{4}(y_{t-1}, y_t, y_{t+1}, y_{t+2}).$$

Moving average over moving average:

$$\begin{aligned}
\mathrm{MA}(\mathrm{MA}(y_t, 4); 2) &= \frac{1}{2}[\mathrm{MA}(y_{t-1}; 4), \mathrm{MA}(y_t; 4)] = \\
&= \frac{1}{2}\left[\frac{1}{4}(y_{t-2}, y_{t-1}, y_t, y_{t+1}) + \frac{1}{4}(y_{t-1}, y_t, y_{t+1}, y_{t+2})\right] = \\
&= \frac{1}{8} y_{t-2} + \frac{1}{4} y_{t-1} + \frac{1}{4} y_t + \frac{1}{4} y_{t+1} + \frac{1}{8} y_{t+2}.
\end{aligned}$$

MAs are used to: 1) smooth out the data; 2) extranct the trend.

Weighted moving average (WMA):

$$\mathrm{WMA}(y_t; m) = \sum_{j=-k}^{k} y_{t+j} \cdot w_j, \;\; w_j \geq 0, \;\; \sum w_j = 1.$$

The classical TSD algorithm is given as follows:

1. Compute trend component using $2 \times m$-MA if m is even and m-MA if it is odd.

$$\hat{T}_t = \begin{cases} \mathrm{MA}(y_t; m), \text{ if } m \text{ is odd}, \\ \mathrm{MA}(\mathrm{MA}(y_t; m); 2), \text{ if } m \text{ is even}. \end{cases}$$

2. Detrend the time series (TS):

$$y_t - \hat{T}_t = S_t + R_t.$$

3. Compute $\hat{S}_t$ by averaging detrended TS for a season (assuming that S_t does not change from season to season).
4. $\hat{R}_t = y_t - \hat{S}_t - \hat{T}_t$.

Note: TSD assumes that S_t is constant throughout the seasons and that the trend line itself is not sensitive to sharp fluctuations.

1. 2. STL decomposition

An alternative to classical TSD would be *STL decomposition* (Seasonal Trend decomposition via LOESS). Here LOESS (locally estimated scatterplot smoothing) is type of local regression for modeling and smoothing data $(x_i, y_i)_{i=1}^{m}$. Its key components are:

1. Kernel function. For example, Gaussian kernel

$$w_i = \exp\left(-\frac{(x_i - x)^2}{2\tau^2}\right).$$

2. Smoothing parameter τ. Smaller τ leads to narrower windows and more flexible models, larger τ — to wider windows and less flexible models and $\tau \to +\infty$ means that $w_i = 1$, hence model becomes a simple linear regression.

Given data $(x_i, y_i)_{t=1}^{m}$ or $(t, y_t)_{t=1}^{T}$, the LOESS algorithm step-by-step:

1. Choose a kernel function $\mathcal{F}$ and set smoothing parameter τ.

2. For all x_i:

   2.1. Calculate $w_i = \mathcal{F}(x_i, x, \tau)$

   2.2. Build weighted regression model. For example, weighted least squares:

$$L = \sum_{i=1}^{n} w_i (y_i - \Theta^T x_i)^2,$$

   where $\Theta = (X^T W X)^{-1} X^T W y$.

   2.3. Make predictions $\hat{y}(x)$ for x only.

   2.4. "Forget" the model.

1. 2. 1. STL algorithm

Input: $Y = \{y_1, ..., y_\tau\}$.

Parameters: n_p — # of outer iterations (1-2)

n_i — # of inner iterations (1-2)

n_l — trend smoothing parameter (smoothing parameter for LOESS)

n_s — seasonality smoothing parameter

n_o — residual smoothing parameter (optional, for residues R_t).

0. Outer loop: repeat the following steps n_p times.
1. Initialization:

   1.1. set trend $T^{(0)} = 0$ or other initial approximation (MA for example);

   1.2. set weights $w = \{1, 1, ...1\}$ (optional, for residues).

2. Inner loop: repeat n_i times

   2.1. Detrend time series: $D = Y - T$.

   2.2. Compute seasonal component:

   2.2.1. Split D subseries by seasons;

   2.2.2. For each subseries apply the LOESS smoothing with $\tau = n_s$ and weights w.

   2.2.3. Assemble the smoothed subseries into a seasonal component C.

2.2.4. Center the seasonal component C by subtracting moving average.

2.3. Update seasonal component $S = C$.

2.4. Deaseasonalize the data: $Y_{\text{desd}} = Y - S$

2.5. Update the trend: apply LOESS for Y_{desd} with $\tau = n_l$ and "robust" weights w (obtain T).

3. Compute the residuals $R = Y - T - S$.
4. Update weights: recompute weights w based on residues R to reduce the influence of outliers usually using Tuikey's biweight function.

Post-processing:

1. Normalize seasonality: mean value of S for each season should be zero.
2. Smoothen the trend if needed.

Result: trend T, seasonality S, residual noise R

Pros:

- *flexiblity:* it is robust to outliers;
- *robustness:* it can model non-linear trends;
- *arbitrary period:* it can work with any seasonality.

1. 2. 2. Tuikey's biweight function

Tuikey's biweight function is used to update the weights w using the following algorithm:

1. Obtain the residuals $R = Y - S - T$
2. Compute MAD (median absolute deviation):

$$\text{MAD} = \text{median}(|r_i - \text{median}(R)|).$$

Normalize: $S \approx 1.4826 \cdot \text{MAD}$, since $\sigma = 1.4826$

3. Compute the normalized residuals:

$$u_i = \frac{r_i}{C \cdot S},$$

where C is a tuning constant ($C = 4.685$).

4. Bisquare function

$$w_i = \begin{cases} (1 - u_i)^2, \ |u_i| < 1, \\ 0, \ |u_i| \geq 1. \end{cases}$$

5. If $S = 0$, then $w_i = 0$ (all residuals are the same). If $\text{MAD} = 0$, but the residuals are not the same, we use standard deviation instead of MAD.

For example, if $R = [0.1, -0.2, 3.0, -0.1, 10.0]$:

1. $\text{median}(R) = 0.1$, hence $\text{MAD} = \text{median}(|R - 0.1|) = 0.3$
2. $S = 0.3 \cdot 1.4826 \approx 0.4448$
3. $C = 4.685 \Rightarrow C \cdot S = 2.083$
4. $r_3 = 3.0 : |u_3| = |\frac{3.0}{2.083}| \approx 1.44 > 1 \Rightarrow u_3 = 0$
5. $r_3 = 10.0 : |u_5| = 4.801 > 1 \Rightarrow u_5 = 0$
6. $r_1 = 0.1 : |u_1| \approx 0.04821 \Rightarrow w_1 \cdot (1 - 0.048^2)^2 \approx 0.995$

2. Weak, strong stationarity. Stationarity tests: DF, ADF, KPSS. Reduction to stationary time series.

2. 1. Stationarity and Ergoticity

Stationarity is a key feature of time series. There are several kinds of stationarity:

- *Strict stationarity:* joint distribution of any segment of time series $\left(y_{t_1}, y_{t_2}, \ldots, y_{t_k}\right)$ is equivalent to $\left(y_{t_1+\tau}, y_{t_2+\tau}, \ldots, y_{t_k+\tau}\right)$ $\forall \tau$.
- *Weak stationarity:*

1. $\forall t \ \mathbb{E}[y_t] = \mu$,

2. $\forall t \ \mathbb{D}[y_t] = \sigma^2 < +\infty$,

3. $\forall t, s, \tau \ \ \operatorname{cov}(y_t, y_s) = \operatorname{cov}(y_{t+\tau}, y_{s+\tau}) = \gamma(|t-s|)$. Here $\gamma(\cdot)$ is a function that depends on distance between points.

2. 1. 1. Non-stationary time series examples

1. Time servies with determinitstic trend:

$$y_t = \alpha + \beta t + \varepsilon_t, \ \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

Here, $\mathbb{E}[y_t] = \alpha + \beta t$ which is not a constant value.

2. $y_t = \sin t + \varepsilon_t, \ \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. Here

$$\mathbb{E}[y_t] = \begin{cases} 1, \ t = \frac{\pi}{2} + 2\pi k \\ -1, \ t = -\frac{\pi}{2} + 2\pi k \end{cases}$$

and since it depends on t the TS is non-stationary.

3. Random Walk: $y_t = y_{t-1} + \varepsilon_t, \ \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \ \ \operatorname{cov}(\varepsilon_t, \varepsilon_s) = 0, \ t \neq s$. Let us write out values of this TS:

$$\begin{aligned} y_1 &= y_0 + \varepsilon_1, \\ y_2 &= y_1 + \varepsilon_2 = y_0 + \varepsilon_1 + \varepsilon_2, \\ &\ldots \\ y_t &= y_0 + \sum_{i=1}^{t} \varepsilon_i \end{aligned}$$

Therefore, $\mathbb{E}[y_t] = y_0, \ \mathbb{D}[y_t] = t\sigma^2$.

2. 1. 2. Stationary time series examples

1. $y_t = \varepsilon_t, \varepsilon_t \underset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ – white noise. In this case,

$$\forall t, s : t \neq s,\ \mathbb{E}[y_t] = 0,\ \mathbb{D}[y_t] = \varepsilon^2 < \infty \to \text{stationary}$$

2. $y_t = \beta_1 y_{t-1} + \varepsilon_t,\ \beta \in (-1, 1),\ \varepsilon_t \underset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$

$$\begin{aligned} y_t &= \beta_1 y_{t-1} + \varepsilon_t = \beta_1(\beta_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t = \\ &= \beta_1^t y_0 + \sum_{i=1}^{t} \beta_1^{t-i} \varepsilon_i. \end{aligned}$$

Here, since ε_i are independant from each other:

$$\begin{aligned} \mathbb{E}[y_t] &= \mathbb{E}\left[\beta_1^t y_0 + \sum_{i=1}^{t} \beta_1^{t-i} \varepsilon_i\right] = \beta_1^t y_0 + \sum_{i=1}^{t} \beta_1^{t-i} \mathbb{E}[\varepsilon_i] = \\ &= \beta_1^t y_0 \quad \text{if} \quad t \to \infty,\ \beta_1^t \to 0. \\ \mathbb{D}[y_t] &= \mathbb{D}\left[\beta_1^t y_0 + \sum_{i=1}^{t} \beta_1^{t-i} \varepsilon_i\right] = \sum_{i=1}^{t} \beta_1^{2(t-i)} \mathbb{D}[\varepsilon_i] = \\ &= \left(\beta_1^{2t-2} + \beta_1^{2t-4} + \ldots + 1\right) \cdot \sigma^2 \\ \operatorname{cov}(y_t,\ y_{t+1}) &= \operatorname{cov}\Big(\beta_1 y_{t-1} + \varepsilon_t,\ \beta_1 y_t + \varepsilon_{t_1}\Big) \\ &= \operatorname{cov}\left(\beta_1^t y_0 + \sum_{i=1}^{t} \beta_1^{t-i} \varepsilon_i,\ \beta_1^{t+1} y_0 + \sum_{i=1}^{t+1} \beta_1^{t+1-i} \varepsilon_i\right) = \\ &= \beta_1 \operatorname{cov}(\varepsilon_t,\ \varepsilon_t) + \beta_1^3 \operatorname{cov}(\varepsilon_{t-1},\ \varepsilon_{t-1}) + \ldots + \beta_1^{2t-1} \operatorname{cov}(\varepsilon_1,\ \varepsilon_1) = \\ &= \sum_{i=1}^{t} \beta_1^{2i-1} \mathbb{D}[\varepsilon_{t+1-i}] \to \frac{\beta_1}{1 - \beta_1^2} \cdot \sigma^2 = \text{const.} \end{aligned}$$

A random stochastic process is called *ergodic* if its statistical properties can be estimated using a sample from it.

Note: any ergodic process is stationary and almost any stationary process is ergodic.

2. 2. Stationarity tests

2. 2. 1. Unit root

Time series with unit root do not have a constant average level and have stochastic trends.

Let us consider a simple model: $y_t = \varphi \cdot y_{t-1} + \varepsilon_t,\ \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2),\ \varphi$ is constant.

1. $|\varphi| < 1$ means that the process is stationary;
2. $|\varphi| > 1$ is a non-stationary or explosive time series;
3. $|\varphi| = 1$ is the unit root case, not stationary, since:

$$y_t = y_{t-1} + \varepsilon_t = y_0 + \sum_{i=1}^{n} \varepsilon_i \Rightarrow \mathbb{D}[y_t] = t\sigma^2.$$

Why unit root?

Let us define a lag operator $Ly_t = y_{t-1}$. Then, $y_t = \varphi y_{t-1} + \varepsilon_t$ can be rewritten as $y_t = \varphi L y_t + \varepsilon_t$ hence $y_t(1 - \varphi L) = \varepsilon_t$.

Taking this into account, the characteristic equation would be

$$(1 - \varphi z) = 0 \Rightarrow z = \frac{1}{\varphi}$$

and if $\varphi = 1$ then $z = 1$ and $y_t = y_{t-1} + \varepsilon_t$.

2. 2. 2. Dickey-Fuller test (unit root test)

1. Consider a time series $y_t = \varphi y_{t-1} + \varepsilon_t$. Let $\Delta y_t = y_t - y_{t-1}$, then:

$$\Delta y_t = (\varphi - 1) y_{t-1} + \varepsilon_t = \gamma y_{t-1} + \varepsilon_t.$$

2. Formulate the hypotheses:

$$H_0 : \gamma = 0\ (\varphi = 1) \Rightarrow \text{unit root} \Rightarrow \text{non-stationary time series.}$$
$$H_1 : \gamma < 0\ (\varphi < 1) \Rightarrow \text{no unit root} \Rightarrow \text{stationary time series.}$$

3. Evaluate γ by fitting regression: $\Delta y_t = \gamma y_{t-1} + \varepsilon_t$. Estimate standard t-statistic for γ:

$$t_{\text{stat}} = \frac{\hat{\gamma}}{\text{SE}(\hat{\gamma})}$$

4. Dickey-Fuller distribution: if H_0 is correct, t_{stat} does not follow the standard t-distribution, it follows Dickey-Fuller distribution.

Significance level	Critical value
1%	-3.43
5%	-2.86
10%	-2.57

5. If $t_{\text{stat}} <$ crit. val. $\to H_0$ is rejected,

If $t_{\text{stat}} >$ crit. val. $\to H_0$ is not rejected.

2. 2. 3. Modification of DF test

Basic regression is very simple model. Instead, it is often epxanded:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \varepsilon_t.$$

This model is able to perform stationarity checks around deterministic trends.

2. 2. 4. Augmented Dickey-Fuller test

DF test assumes that ε_t are not correlated. This issue can be solved by adding lagged differences to the regression. Those lagged differences will reduce autocorrelation in error terms ε_t.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i}$$

How does the choice of p impact the model:

- if p is too small, then the correlation issue will not be solved,
- if p is too big, then the power of test decreases.

How to choose p:

1. $p \approx \sqrt[3]{T},\ p \approx \sqrt{T}$.
2. Test different p, choose p which gives you the “best” regression: BIC, AIC, MQIC.

Interpretation of ADF is exactly the same.

2. 2. 5. KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test

1. KPSS assumes that the time series can be decomposed into the following sum:

$$y_t = \xi_t + r_t + \varepsilon_t,$$

where:

- ξ_t is deterministic trend,
- r_t is stochastic trend such that $\mathbb{D}[r_t] = \sigma_r^2$,
- ε_t – white noise.

2. H_0: time series is stationary $\Rightarrow \sigma_r^2 = 0 \Rightarrow y_t = \xi_t + \varepsilon_t$,

H_1: time series is not stationary $\Rightarrow \sigma_r^2 > 0 \Rightarrow r_t \neq 0$.

3. Fit regression:

   3.1. $y_t = \alpha + \beta t + \varepsilon_t \Rightarrow$ residuals $e_t = y_t - \hat{\alpha} - \hat{\beta} t$.

   3.2. Accumulation of residuals $S_t = \sum_{i=1}^{t} e_i$.

   3.3. Calculate KPSS value:

$$\mathrm{KPSS} = \sum_{i=1}^{T} \frac{S_t^2}{T^2 \sigma_\varepsilon^2},$$

   where σ_ε^2 is the variance of ε_t estimated using Newey-West method.

4. Decision logic: if $\mathrm{KPSS} < \mathrm{crit.\ value}$, reject H_0. Otherwise, H_0 is not rejected.

3. Filtration problem. Deterministic methods of filtration: MA, SMA, EMA, polynomial smoothing.

3. 1. Main methods of reduction to stationary time series

There are tow types of non-stationarity:

1. Trend
2. Nonconsistent dispersion

If there is a trend, we can use the following methods to standardize the time series:

1. Taking difference of time series:

$$y_i \to \Delta y_i, \ \Delta y_i = y_i - y_{i-1}, \ i = 2, ..., \tau.$$

2. Substracting the trend component:

   2.1. TSD $\to$ Trend $\to$ y_i $-$ Trend;

   2.2. Polynomial regression.

2.* Lagged difference:

$$y_i \to \Delta_k y_i, \ \Delta_k y_i = y_i - y_{i-k}$$

and adjust k for seasonality.

2.** Subtract the seasonal component:

$$\text{TSD} \to \text{Seasonal component} \to y_i - \text{Season}$$

3. 1. 1. Dispersion stabilization.

1. Box-cox transformation. Given $y = \{y_1, ..., y_\tau\}, \ y_i > 0$:

$$\tilde{y}_i = \begin{cases} \frac{y_i^\lambda - 1}{\lambda}, \ \lambda \neq 0, \\ \log y_i, \ \lambda = 0. \end{cases}$$

Note: if $\lambda > 1$ the inverse transform is taken, otherwise:

$$\lambda = \begin{cases} 1 \Rightarrow \text{no transformation;} \\ 0.5 \Rightarrow \text{square root, i.e. softer than log;} \\ 0 \Rightarrow \text{natural} \quad \log . \end{cases}$$

The λ value is chosen using a maximum likelyhood function by applying Box-Cox for different λ values and choosing which maximizes the likelyhood of transformed data following a normal distribution.

Normal distribution likelyhood function:

$$L = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z_i - \mu)^2}{2\sigma^2}\right).$$

Substituting $z_i = \tilde{y}_i = \text{Box-Cox}(y_i, \lambda)$ we get:

$$L = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\tilde{y}_i - \mu)^2}{2\sigma^2}\right) \times \prod_{i=1}^{n} y_i^{\lambda-1}$$

$$\log L = -\frac{n}{2}\log\pi - \frac{n}{2}\log\sigma^2 - \sum_{i=1}^{n}\log\left(-\frac{(\tilde{y}_i - \mu)^2}{2\sigma^2}\right) + \\ +(\lambda-1)\sum_{i=1}^{n}\log y_i .$$

Here, the term

$$\prod_{i=1}^{n} y_i^{\lambda-1}$$

is derivative of Jacobian matrix of Box-Cox trnasform. Note that Box-Cox works only for positive y_i, hence if $y_i \leq 0$, the data is shifted by $\alpha : y_i + \alpha > 0,\ i = 1, ...\tau$ and the transform itself is applied after that.

When to apply Box-Cox:

1. Graphical test: plot variance against mean. Use Box-Cox if there is a clear dependance.
2. Distribution is asymmetric.

3. 2. Autocorrelation and partial autocorrelation.

ACF *(AutoCorrelation Function)* shows correlation of y_t with lagged component of time series y_{t-k} for different k's. It is given by the following expression:

$$\mathrm{ACF}(k) = \rho(y_t,\ y_{t-k}) = \frac{\mathrm{cov}(y_t,\ y_{t-k})}{\sigma(y_t)\sigma(y_{t-k})} \approx \frac{\sum_{\tau=k}^{T}(y_k - \overline{y})(y_{t-k} - \overline{y})}{\sum_{t=1}^{T}(y_t - \overline{y})},$$

where $\overline{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$ and $|\mathrm{ACF}(k)| \leq 1$.

ACF is used to identify:

1. **Trend.** Since trend is a long-term movement in a set direction, ACF will be positive and significant for long periods of time.
2. **Memory of the process.** Memory of the process is extent of the effect that previous values have on new observations. Therefore, the rate and nature of autocorrelation attenuation can signify the type of process: if it is fast, i.e. there are drops, the process has short memory; if attenuation is slow, i.e. the changes are exponential, the process has long memory.
3. **Seasonality.** Since seasonality is just oscillations at a fixed frequency, ACF plot will show spikes corresponding to seasonality period.

PACF *(Partial AutoCorrelation Function)* shows correlation between y_t and y_{t-k} but removes the effect of all intermediate lags $(y_{t-1}, y_{t-2}, \dots, y_{t-k+1})$.

$$\mathrm{PACF}(k) = \rho(y_t,\ y_{t-k} |\ y_{t-1}, \dots, y_{t-k+1}).$$

PACF is calculated by fitting a regression

$$y_t = \varphi_{k_1} y_{t-1} + \varphi_{k_2} y_{t-2} + \dots + \varphi_{k_k} y_{t-k} + \varepsilon_t$$

and then $\varphi_{k_k} = \mathrm{PACF}(k)$. Here the terms $\varphi_{k_1}, \dots, \varphi_{k_{k-1}}$ are responsible for removal of linear effect of intermediate lags.

Linear models we may look up: AR(k), MA(k), ARMA(p, k), ARIMA(k).

3. 3. Data filtration and smooting

Data filtration is **not** smooting. Rather smoothing is a tool used in data filtration. Filtration is time series transformation aimed at highlinghting, analyzing or supressing certain characterstics of time series such as noise or artifacts.

Goals of filtration:

- Trend extraction;
- Noise supression;
- Artifact removal;
- Time series decomposition.

A problem that may arise during filtering is finding a compromise between precision and smoothing.

3. 3. 1. Deterministic methods of filtration

1. SMA (moving average):

$$\mathrm{SMA}(y_t, m) = \frac{y_{t-m} + y_{t-m+1} + \ldots + y_{t+m}}{2m+1}.$$

Here the issue arises from last m observations missing, hence in most scenarios the formula will look like this:

$$\mathrm{SMA}\ (y_t, m) = \frac{y_{t-m} + y_{t-m+1} + \ldots + y_t}{m+1}.$$

2. WMA (weighted moving average):

$$\mathrm{WMA} = \frac{\sum w_i y_i}{\sum w_i}.$$

3. EMA (exponential moving average).

The idea of this method is to construct a recurrent formula so that the weights of previous points would decrease exponentially. It is given by the following expression:

$$\mathrm{EMA}(y_t) = \alpha y_t + (1-\alpha)\ \mathrm{EMA}(y_{t-1}).$$

Here $\alpha \in (0,\ 1)$ is a smooting parameter.

4. Polynomial (Savitzky-Golay) filter.

Given data points, choose a window of size $n = 2m+1$ and fit a polynomial line of a low degree then choose its value at i as TS value at i. Algorithm step-by-step (at point i):

1. Choose the window of size $n = 2m+1$.
2. Fit a polinomial $P(i) = \alpha_0 + \alpha_1 i + \alpha_2 i^2 + \ldots + \alpha_k i^k,\ i = -m, \ldots, m.$

3. Least squares minimization:

$$\sum_{i=-m}^{m} (P(i) - y_i)^2 \to \min_{\alpha_j}$$

4. $P(0) = \hat{\alpha}_0 \to$ smoothed value for current y_t.

Downside: polynomials fitted for each point, which is suboptimal.

$\hat{\alpha}_0$ can be expressed as weighted combination of all y_i inside the window:

$$\hat{\alpha}_0 = c_{-m} y_{-m} + c_{-m+1} y_{-m+1} + ... + c_m y_m,$$

where c_j are coefficients of Savitzky-Golay filter, which depend on window size and degree of polynomial.

How to compute c_j:

1. $P(i) = \alpha_0 + \alpha_1 i + ... + \alpha_k i^k$

2.

$$\begin{aligned} P(-m) &= \alpha_0 + \alpha_1 \cdot (-m) + ... + \alpha_k \cdot (-m)^k \approx y_{-m}, \\ &... \\ P(0) &= \alpha_0, \\ &... \\ P(m) &= \alpha_0 + \alpha_1 m + ... \alpha_k m^k. \end{aligned}$$

In matrix multiplication from:

$$X\alpha \approx y.$$

Here,

$$X = \begin{pmatrix} 1 & -m & (-m)^2 & ... & (-m)^k \\ 1 & -m+1 & (-m+1)^2 & ... & (-m+1)^k \\ ... & ... & ... & ... & ... \\ 1 & m & m^2 & ... & m^k \end{pmatrix}$$

and α is target for linear regression

$$\|X\alpha - y\|^2 \to \min_{\alpha}.$$

Taking its solution we get $\hat{\alpha} = \left(X^T X\right)^{-1} X^T y$

$$\hat{\alpha}_0 = c_0^T \hat{\alpha} = c_0^T \left(X^T X\right)^{-1} X^T y, \;\; c_0 = [1, 0, ..., 0]^T$$
$$\hat{\alpha}_0 = C^T y = c_{-m} y_{-m} + ... + c_m y_m.$$

How to deal with corner points:

1. Asymmetric window
2. Use polynomials calculated for the first and last full window.

Derivatives of the signal:

$$P'(i) \mid_{i=0} = \hat{\alpha}_1,$$
$$P''(i) \mid_{i=0} = 2\hat{\alpha}_2,$$
$$P^{(n)}(i) \mid_{i=0} = n!\hat{\alpha}_n.$$

4. Filtration and smoothing using Fourier analysis: Fourier series, Fourier transform, DFT, FFT, SSFT.

4. 1. Fourier transform

Fourier series is a decomposition of a function $f \in C[a,b]$ with a orthogonal function system $\{g_k\}_{k=0}^{+\infty}$ in some euclidean space:

$$f(x) = \sum_{k=1}^{+\infty} c_k g_{k(x)},\ (f, g_k) = \int_a^b f(x) g_k(x) dx = 0$$

If g_k is a trigonometric system:

$$g_k \in \left\{ \frac{1}{2l}, \frac{1}{\sqrt{l}} \cos\left(\frac{\pi x}{l}\right), \frac{1}{\sqrt{l}} \sin\left(\frac{\pi x}{l}\right), \ldots \right\}$$

Then $f(x)$:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} \left[a_k \cos\left(\frac{k\pi x}{l}\right) + b_k \sin\left(\frac{k\pi x}{l}\right) \right],$$

$$a_k = \frac{1}{l} \int_{-l}^{l} f(x) \cos\left(\frac{k\pi x}{l}\right) dx,\ a_{-k} = a_k,$$

$$b_k = \frac{1}{l} \int_{-l}^{l} f(x) \sin\left(\frac{k\pi x}{l}\right) dx,\ b_0 = 0,\ b_{-k} = -b_k.$$

In a more general case:

$$f(x) = \sum_{k=-\infty}^{+\infty} c_k e^{i w_k x},\ w_k = \frac{\pi k}{l},\ c_k = \frac{1}{2l} \int_{-l}^{l} f(x) e^{-i w_k x} dx$$

Let us derive this statement. Since

$$\sin(kx) = \frac{e^{ikx} - e^{-ikx}}{2i} \ \text{ and } \ \cos(kx) = \frac{e^{ikx} + e^{-ikx}}{2},$$

$f(x)$ can expressed in a following manner:

$$f(x) = e^{iw_0x} \cdot \frac{a_0}{2} + \sum_{k=1}^{+\infty}\left[a_k\frac{e^{iw_kx} + e^{-iw_kx}}{2} + b_k\frac{e^{iw_kx} - e^{-iw_kx}}{2i}\right] =$$

$$= \frac{a_0}{2}e^{iw_0x} + \frac{1}{2}\sum_{k=1}^{+\infty}[a_ke^{iw_kx} + a_ke^{-iw_kx} - ib_ke^{iw_kx} + ib_ke^{-iw_kx}] =$$

$$= \frac{a_0}{2}e^{iw_0x} + \frac{1}{2}\sum_{k=1}^{+\infty}(a_k - ib_k)e^{iw_kx} + \frac{1}{2}\sum_{k=1}^{+\infty}(a_k + ib_k)e^{-iw_kx} =$$

$$= \sum_{k=-\infty}^{+\infty} c_ke^{iw_kx}.$$

Then, since $a_{-k} = a_k$ and $b_{-k} = -b_k$,

$$c_k = \frac{1}{2}(a_k - ib_k) = \frac{1}{2l}\int_{-l}^{l} f(t)\left(\cos\left(\frac{k\pi t}{l}\right) - i\sin\left(\frac{k\pi t}{l}\right)\right)dt =$$

$$= \frac{1}{2l}\int_{-l}^{l} f(t)\left(\frac{e^{iw_kt} + e^{-iw_kt}}{2} - i\frac{e^{iw_kt} - e^{-iw_kt}}{2i}\right)dt =$$

$$= \frac{1}{4l}\int_{-l}^{l} f(t) \cdot 2e^{-iw_kt}dt = \frac{1}{2l}\int_{-l}^{l} f(t)e^{-iw_kt}dt.$$

4. 2. From Fourier series to Fourier transform

For $t \in [-l, l]$:

$$f(t) = \sum_{k=-\infty}^{+\infty} c_ke^{iw_kt}.$$

However, for $l \to +\infty$ following assumptions should be made:

1. $f(t)$ is piecewise continuous and has one-sided derivative in $[-l, l]$.
2. Limit function $f(t) = \lim_{l\to+\infty}\sum_{k=-\infty}^{+\infty} c_ke^{iw_kt}$ is absolutely integrable.
3. Limit function $f(t)$ is piecewise continuous and has one-sided derivatives at any point.

Let us define $\Delta w_k = w_{k+1} - w_k,\ k \in \mathbb{Z}$. Since $w_k = \frac{\pi k}{l}$, $\Delta w_k = \frac{\pi}{l}$ and $\frac{1}{l} = \frac{\Delta w_k}{\pi}$. Therefore, $f(t)$ can be represented as:

$$
\begin{aligned}
f(t) = & \sum_{k=-\infty}^{+\infty} \frac{1}{2l} \int_{-l}^{l} f(\tau) e^{-iw_k\tau} d\tau \cdot e^{iw_k t} = \\
= & \sum_{k=-\infty}^{+\infty} \frac{1}{2\pi} \int_{-l}^{l} f(\tau) e^{-iw_k\tau} d\tau \cdot e^{iw_k t} \Delta w_k = \\
= & \frac{1}{2\pi} \sum_{k=-\infty}^{+\infty} \hat{F}_l(w_k, t) \Delta w_k.
\end{aligned}
$$

And if $l \to +\infty$, then $\Delta w_k \to 0$ and

$$
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{F}_l(w, t) dw,
$$

where $\hat{F}_l(w, t) = \int_{-\infty}^{+\infty} f(\tau) e^{-iw\tau} d\tau \cdot e^{iwt}$.

Then, **Fourier transform** can be defined as:

$$
\hat{f}(w) = \mathcal{F}(f(t)) = \int_{-\infty}^{+\infty} f(t) e^{-iwt} dt.
$$

And **inverse Fourier transform** would be:

$$
f(t) = \mathcal{F}^{-1}\left(\hat{f}(w)\right) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(w) e^{iwt} dw.
$$

Properties of Fourier transform:

1. Linearity.
2. $\mathcal{F}(f * g) = \hat{f}(w) \cdot \hat{g}(w)$, where $f * g = \int_{-\infty}^{+\infty} f(\tau) g(t - \tau) d\tau$ i.e. convolution.
3. $\mathcal{F}(f \cdot g) = \hat{f}(w) + \hat{g}(w)$.

4. 3. Discrete Fourier transform

DFT (discrete Fourier transform) is an operation that transforms $f(t)$ to $f_0, f_1, ..., f_n$. Direct and inverse DFT respectively:

$$\hat{f}_k = \sum_{j=0}^{n-1} f_j \exp\left(-i\frac{2\pi jk}{n}\right),$$
$$f_k = \sum_{j=0}^{n-1} \hat{f}_j \exp\left(i\frac{2\pi jk}{n}\right).$$

It has algorithmic complexity of $\mathcal{O}(n^2)$ and is essentially a matrix multiplication:

$$\begin{pmatrix} \hat{f}_0 \\ \vdots \\ \vdots \\ \vdots \\ \hat{f}_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w_n & w_n^2 & \dots & w_n^{n-1} \\ 1 & w_n^2 & w_n^4 & \dots & w_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_n^{n-1} & w_n^{2(n-1)} & \dots & w_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} f_0 \\ \vdots \\ \vdots \\ \vdots \\ f_{n-1} \end{pmatrix}$$

where $w_n = \exp\left(-\frac{2\pi i}{n}\right)$.

4. 4. Fast Fourier transform

FFT (fast Fourier transform) is a family of algorithms that arose from need for a… faster version of DFT. Let us consider Cooley-Tukey algorithm. It relies on two properties of DFT:

- $w_n^{jk} = \exp\left(-i\frac{2\pi jk}{n}\right)$ is periodic: $w_n^{jk} = w_n^{j(k+n)} = w_n^{k(j+n)}$.
- w_n^{jk} is symmetric: $w_n^{k+\frac{n}{2}} = -w_n^k$.

The algorithm step-by-step:

1. Split f into even and odd terms: $f_{\text{even}} = \{f_{2k}\}_{k=0}^{\frac{n}{2}-1}$ and $f_{\text{odd}} = \{f_{2k+1}\}_{k=0}^{\frac{n}{2}-1}$
2. Let $G(k) = \text{DFT}(f_{\text{even}})$ and $H(k) = \text{DFT}(f_{\text{odd}})$ which takes $\mathcal{O}\left(\frac{n^2}{4}\right)$ operations each and $\mathcal{O}\left(\frac{n^2}{2}\right)$ total.

Therefore,

$$\hat{f}_k = \sum_{j=0}^{\frac{n}{2}-1} f_{2j} \exp\left(-i\frac{2\pi k(2j)}{n}\right) + \sum_{j=0}^{\frac{n}{2}-1} f_{2j+1} \exp\left(-i\frac{2\pi k(2j+1)}{n}\right) =$$
$$= G(k) + w_n^k H(k),\ \ k = 0, 1, \dots, \frac{n}{2} - 1.$$

Taking the periodicity of w_n into account,

$$\hat{f}_{k+\frac{n}{2}} = G\left(k + \frac{n}{2}\right) + w_n^{k+\frac{n}{2}} H\left(k + \frac{n}{2}\right) = G(k) - w_n^k H(k)$$

which implies that $H(k+\frac{n}{2}) = H(k)$ and $G(k+\frac{n}{2}) = G(k)$. This implies that for $k \in \{\frac{n}{2}, ..., n-1\}$ $\hat{f}_k$ can be calculated using the values from a period before:

$$f_{k+\frac{n}{2}} = G\left(k+\frac{n}{2}\right) + w_n^{k+\frac{n}{2}} H\left(k+\frac{n}{2}\right) = G(k) - w_n^k H(k),\ k = 0, ..., \frac{n}{2} - 1.$$

3. Recursion. It can be used to calculate $H(k)$ and $G(k)$, moreover, when $n = 2^m$ recursion can be applied untill the end.

FFT complexity. Total number of recursions is $m = \log_2 n$, hence it is $\mathcal{O}(n \log_2 n)$.

Matrix form.

$$\hat{f} = F^{2^m} f = \begin{pmatrix} E^{2^{m-1}} & D^{2^{m-1}} \\ D^{2^{m-1}} & E^{2^{m-1}} \end{pmatrix} \begin{pmatrix} F^{2^{m-1}} & 0 \\ 0 & F^{2^{m-1}} \end{pmatrix} \begin{pmatrix} f_{\text{even}} \\ f_{\text{odd}} \end{pmatrix},$$

where E^n is $n \times n$ identity matrix and

$$D^n = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & w_n & 0 & ... & 0 \\ 0 & 0 & w_n^2 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & ... & ... & ... & w_n^n \end{pmatrix}.$$

4. 5. Short time Fourier transform

STFT (Gabor transform) is given by:

$$G(f)(t, w) = \hat{f}_g(t, w) = \int_{-\infty}^{+\infty} f(\tau) e^{-iw\tau} g(\tau - t) d\tau,$$

where $g(t) = \exp\left(-\frac{(t-\tau)^2}{\alpha^2}\right)$ is a Gaussian kernel function, but it is not necessary to use specifically this kernel function.

STFT can easily be discretized by applying FFT in eahc window. The result of STFT is a spectrogram: a plot of frequency against time.

5. Time series forecasting problem. Multi-step ahead forecasting: two main approaches.

6. Exponential smoothing, Holt's linear model, ETS models.

7. Autocorrelation and partial autocorrelation. AR, MA, ARMA, ARIMA models.

8. Predictive clustering

9. Predictive clustering for trajectory forecasting

10. Clusterization for time series: DBSCAN, Wishart, metrics

11. Time series forecasting with neural networks