**Analysis of Sequential Data**

# 1 Stochastic Processes

A stochastic process (SP) is a sequence of random variables indexed in time order: $Y_1, Y_2, Y_3, ..., Y_t$. Realization of a stochastic process yield a time series. Taking a sample from the stochastic process means sampling a possible future behaviour of the system. Using only past data to predict a future is hard, which is why a model of the system behaviour is chosen first. We then try to predict a new value given past observations as $Y_t|Y_{t-1} = Y_t|Y_1, Y_2, ...Y_{t-1}$. Time series can exhibit three major patterns:

Trend: pattern exists when there is a long-term increase or decrease in the data.

Seasonal: pattern exists when a series is influenced by seasonal factors in a predictable and fixed way.

Cyclic: pattern exists when data exhibit rises and falls that are not of fixed period

# 2 Autocorrelation

(Linear) Correlation of random variables of a time series at different times: $AutoCorr(t_1, t_2) = Cor(X_{t1}, X_{t_2})$. Plotting the autocorrelation for different time lags produces a correlogram or Autocorrelation Function (ACF). Seasonality can be seen by high positive values at lags that are multiples of the seasonality; a trend can be seen by higher absolute values for small lags. A time series with both patterns will show a mix of both in the ACF. White noise series will produce values below the critical threshold of $\pm 1.96/\sqrt{T}$ for all lags.

# 3 Statistical Hypothesis Testing (SHT)

SHT is toolbox to decide if given data supports a particular given hypothesis. It works by choosing a null hypothesis, e.g. the time series is white noise. We then choose a statistical test, e.g. Ljung-Box and compute its test statistics for a given signifance level, e.g. 5%. If the p-Value is very small, that means it is very unlikely to sample a series this extreme given the null hypothesis. We thus reject the hypothesis.

# 4 Forecasting

Computation of forecast at different points in time with $\hat{y}_{t|t-1}$ as the fitted value of $y$ at time $t$ given past values up to time $t-1$. We can perform $h-$ step forecasts by iteratively applying one step forecasts. Often, a model is chosen that is well suited to the characteristics of the time series and then fitted to it. This model can then be used to perform forecasts.

## 4.1 Simple Forecast Methods

Average: Forecast of all future values is equal to mean of historical data.

Naïve: Forecast of all future values is equal to the last observed value.

Seasonal Naïve : Forecast of all future values is equal to the last observed value from the same season.

Drift: Forecast equal to last value plus average change in historical values. Results in a straight line between last observed and last forecasted values.

## 4.2 Residual Analysis

Residuals in forecasting are defined as the difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$. Residuals should have zero mean. If they do not, the forecast performed with the fitted model is biased. The should also be uncorrelated. If they are not, the residuals contain information not captured by the fitted model. Residuals are also assumed to have constant variance and to be normally distributed.

## 4.3 Evaluating Forecast Accuracy

The residuals produced by fitting a model should be indistinguishable from white noise by fulfilling all criteria above. We can use SHT with a test like Ljung Box to test whether this is true. Small p-values lead to rejecting the null hypothesis meaning residuals are correlated. Having small residuals does not guarantee good forecasting performance.

Forecast Errors are also used for evaluating forecasts. The data is split into a training and a test set. The training set is used for fitting the chosen model while the test set is used to evaluate its performance. Based on the test set containing $n$ values, different error metrics can be computed: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, $MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$, $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$, $MASE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}$, $SSE = \sum_{t=1}^{n}(y_t - \hat{y}_t)^2$. $MAE$, $MSE$, $RMSE$ and $SSE$ are all scale dependent. $MAPE$ is scale independent but is only sensible if $y_t \gg 0$ for all $t$, and $y$ has a natural zero.

Cross-validation for time series involves iteratively splitting a time-ordered dataset into training and test sets, ensuring that the test set comes after the corresponding training set, to assess the performance of a model on multiple folds and mitigate the risk of temporal information leakage (giving the model access to the future it is supposed to predict).

## 4.4 Prediction Intervals

A prediction interval gives a region within which we expect the forecast value with a specific probability. It serves as a measure of confidence in the prediction. The 95% prediction interval for an $h$-step forecast can be computed as $\hat{y}_{T+h|T} \pm 1.96\hat{\sigma}_h$, with $\hat{\sigma}_h$ as the stddev of the $h$-step distribution. Assuming the residuals are normally distributed, $\hat{\sigma}_h$ can be estimated for different forecast methods as:

Mean: $\hat{\sigma} \cdot \sqrt{1 + 1/T}$, Naïve: $\hat{\sigma} \cdot \sqrt{h}$, Seasonal Naïve: $\hat{\sigma} \cdot \sqrt{k+1}$, Drift: $\hat{\sigma} \cdot \sqrt{h \cdot (1 + h/T)}$ where $\hat{\sigma}$ is the stddev of the residuals, $T$ is the number of historical values, $h$ is the integer part of $(h-1)/m$ and $m$ is the seasonal period.

# 5 Exponential Smoothing

Exponential smoothing is a time series forecasting method that assigns exponentially decreasing weights to past observations, providing a weighted average of historical data to generate a smoothed forecast with a focus on recent trends. Parameters such as $\alpha$ or $\phi$ can be found via optimization of the $SSE$.

## 5.1 Simple Exponential Smoothing (SES)

A a weighted moving average, whose weights decrease exponentially: $\hat{y}_t = \alpha \cdot y_{t-1} + \alpha \cdot (1-\alpha) \cdot \hat{y}_{t-1} + \alpha \cdot (1-\alpha)^2 \cdot \hat{y}_{t-2} + ...$ where $0 \leq \alpha \leq 1$. The weights sum up to one as a geometric series. SES provides flat forecasts for all time horizons as it does not incorporate any trend or cyclic components.

Forecast Equation: $\hat{y}_{t+h|t} = \ell_t$; Smoothing Equation: $\ell_t = \alpha \cdot y_t + (1-\alpha) \cdot \ell_{t-1}$

## 5.2 Trend Methods

**Holt's linear trend** Forecast Equation: $\hat{y}_{t+h|t} = \ell_t + h \cdot b_{t-1}$; Smoothing Equation: $\ell_t = \alpha \cdot y_t + (1-\alpha) \cdot (\ell_{t-1} + b_{t-1})$; Trend Equation: $b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot b_{t-1}$

**Holt's linear with damped trend** Forecast Equation: $\hat{y}_{t+h|t} = \ell_t + h \cdot \phi \cdot b_{t-1}$; Smoothing Equation: $\ell_t = \alpha \cdot y_t + (1-\alpha) \cdot (\ell_{t-1} + \phi \cdot b_{t-1})$; Trend Equation: $b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot \phi \cdot b_{t-1}$

## 5.3 Seasonal Methods

Holt-Winters is an extension to Holt's method to capture seasonality.

**Additive:** seasonality component is constant

Forecast Equation: $\hat{y}_{t+h|t} = \ell_t + hb_{t-1} + s_{t-m+h-m(q+1)}$; Smoothing Equation: $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$; Trend Equation: $b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$; Seasonality Equation: $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$;

**Multiplicative**: seasonality component is proportional to series level

Forecast Equation: $\hat{y}_{t+h|t} = (\ell_t + hb_{t-1}) \cdot s_{t-m+h-m(q+1)}$; Smoothing Equation: $\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1})$; Trend Equation: $b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$; Seasonality Equation: $s_t = \gamma \frac{y_t}{\ell_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m}$;

## 5.4 Exponential Smoothing State Space Models (ETS)

Contrary to pure forecasting methods, State Space Models can provide point forecasts as well as prediction intervals. They assume three components that make up the time series:

Error: Additive, Multiplicative

Trend: None, Additive, Additive damped

Sesonal: None, Additive, Multiplicative

Automatic Forecasting aims to find the best performing model out of all possible permutations above by choosing the model with the lowest $AIC = -2\ln(\mathcal{L}) + 2k$ or $AICc = AIC + \frac{2k(k+1)}{n-k-1}$, where $\mathcal{L}$ is the likelihood of the model, $k$ is the number of estimated parameters in the model and $n$ is the number of observations.

# 6 Autoregressive Integrated Moving Average (ARIMA)

Combines autoregressive (AR) and moving average (MA) components along with differencing (I).

## 6.1 Random Walk (with drift)

$y_t = c + y_{t-1} + \epsilon_t$

## 6.2 Stationarity

Refers to a time series property where statistical properties, such as mean and variance, remain constant over time, providing a stable and predictable behavior. Stationary processes drop quickly to zero and no seasonality visible in the ACF. Non-stationary processes decrease more slowly or not at all.

## 6.3 Differencing

Change between observations in the time series. Can be done multiple times or for observations in the same season. Differencing can help with obtaining a stationary time series. Finding the correct number of differences to obtain stationarity can be automated. Backshift notation: $By_t = y_{t-1}$; $y'_t = y_t - y_{t-1} = (1-B) \cdot y_t$; $y''_t = (1-B)^2 \cdot y_t$

## 6.4 Autoregressive (AR) and Moving Average (MA) models

Autoregressive Models use multiple regression with lagged values of $y_t$ as predictors: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$, denoted as AR(p)

They work only on stationary data. All $\phi$ values are restricted such that the complex roots of the characteristic polynom lie outside the unit circle.

Moving Average Models use multiple regression with past noise as predictors: $y_t = c + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$, denoted as MA(p)

Any $MA(q)$ process can be written as an $AR(\infty)$ when the roots of the characteristic polynom lie outside the unit circle.

## 6.5 ARMA and ARIMA models

ARMA: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots - \theta_q \varepsilon_{t-q}$

ARIMA: Combine ARMA with differencing; p = order of the autoregressive part; d = degree of first differencing involved; q = order of the moving average part

# 7 Digital Signals

Very common type of sequential data that results from sampling an analog signal (A/D Conversion). Sampling frequency $f_s = \frac{1}{T_s}$, where $T_s$ is the sampling period in [s]. Sampling frequency must be at least double the highest present frequency in the analog signal, else aliasing occurs. Normalized frequencies and normalized sampling periods are expressed in cycles per sample and samples per circle and are calculated by dividing the analog by the digital frequency/sampling period.

## 7.1 Discrete Fourier Spectrum

Every periodic signal can be decomposed into its frequency components and their respective amplitudes. This transformation transfers the signal from the time to the frequency domain. Transforming the other way can be done via an Inverse Discrete Fourier Transform.

The spectrum of a signal shows additional frequencies mirrored at $\frac{f_s}{2}$ due to sampling and eulers formula. The amplitudes of a spectrum can also be expressed in dB.

## 7.2 Linear Time-Invariant Systems

Class of signal transformations that map input sequences to output sequences such that the output response is a scaled and time-shifted version of the input, and the system characteristics remain constant over time. Examples are a delay system or a moving average.

LTI Systems outputs can be calculated by convolving the input $\mathbf{x}$ with the impulse response of the system $\mathbf{h}$ as $y_n = \sum_{k=-\infty}^{\infty} x_k \cdot h_{n-k}$.

the frequency response $H(\omega)$ provides a mapping of how the amplitude and phase of different sinusoidal components in an input signal are affected by the LTI system.

Filters Attenuate or amplify certain frequencies of the input signal. Examples include high- and lowpass filters.

# 8 Speech Signals

One of the most challenging temporal signals to analyse.

Phonemes are the smallest sound units that distinguish words. While phones are the concrete, physical sounds produced in actual speech, including their variations and nuances and are therefore speaker dependent.

Fundamental frequencies of speech signals range from 50 Hz (deep mans voice) to 400Hz (child).

## 8.1 Short Time Speech Analysis

As the main information of a speech signal lies in the change of the spectral characteristics over time, we use a shifting window approach and calculate the spectrum for every window.

Analyzing only a window of the signal is equivalent to multiplying the signal with a rectangular window function. Multiplying the window with signal in the time domain means convolution of the spectra of the two signals in the frequency domain.

Spectograms show the temporal change of the frequencies by contouring high energy frequencies darker for all windows of the speech signal. Formants are high energy areas in the spectograms.

## 8.2 Speech Recognition

Speech recognition is the technology that converts spoken language into written text, allowing computers to understand and process verbal commands or transcribe spoken words. Very difficult because human voices can be very different, humans make mistakes, background noises and different acoustics for different environments. Also, there are no word boundaries in human speech.

### 8.2.1 Feature Extraction

For speech recognition, spectograms and the formants are important features. Unimportant information can be filtered out by many means, e.g. DFT-Cepstrum to extract the source (vocal excitation) from the filter (vocal tract) components. A Mel-Spectrum is a spectrum that more closely resembles the characteristics of the human ear. Mel-Cepstrum: IDFT of Log(Mel-Spectrum). The Fast Cochlea Transform (FCT) is a signal processing technique inspired by the auditory processing in the human cochlea.

### 8.2.2 Classical Approaches: Rule- based and Pattern Matching

Both require feature extraction from the speech singal.

Rule- based: Classification is based on rules for each class derived from human knowledge/observation. Very difficult because not generalizeable for different speakers.

Pattern Matching: Classification based on a distortion measure between a feature pattern and given reference patterns for each class (kNN, SVMs). Main problem is that the patterns can also vary in the temporal structure. We can use Dynamic Time Warping (DTW) to align the extracted features to the reference features by locally stretching or squeezing pattern by duplicating or dropping feature vectors.

### 8.2.3 Statistical Classification

To calculate joined probabilities, we can use bayes rule: $P(X|W) = \frac{P(W|X) \cdot P(W)}{P(X)}$

Given extracted feature $X$ from the speech recognizer, the MAP classifier must choose the word sequence $W$ that has the highest a-posteriori probability of all possible word sequences.

### 8.2.4 Hidden Markov Model (HMM)

Defined by N states, state transition probabilities as well observation probability distribution in each emitting state.

Forward Algorithm: Given a HMM $\lambda$, a sequence of emitted observations $\mathbf{X} = x_1, x_2, ..., x_T$, we want to efficiently calculate $P(\mathbf{X}|\lambda)$. Known as Evaluation problem. Brute force approach would be to simply try out all state combinations and see which one has the highest likelihood of giving the seen sequence. Forward algorithm takes advantage of independent states and instead keeps the probabilities for emitting the observations for each state. Forward: Addition.

Viterbi Algorithm: Given a HMM $\lambda$, a sequence of emitted observations $\mathbf{X} = x_1, x_2, ..., x_T$, we want to efficiently calculate the most likely state sequence $Q* = \max_Q P(Q|\mathbf{X}, \lambda)$. Known as Decoding Problem. Same idea as Forward, but instead of adding all possibilities, we always take the max. N-best Viterbi outputs the N-best state sequences instead of only the best.

Forward-Backward algorithm/Baum-Welch algorithm: Given a sequence of emitted observations $\mathbf{X} = x_1, x_2, ..., x_T$ and a model structure, find parameters for HMM $\omega$ such that $\omega* = max_\lambda P(\mathbf{X}|\lambda)$. Known as Estimation Problem. How HMMs are trained. Forward-Backward estimates the probabilities of the sequence given the current models while Baum-Welch applies changes to optimize the parameters, kind of like backpropagation.

### 8.2.5 HMM Speech Recognizer

Speech recognition is done with HMMs of sub-units that are concatenated to word and sentence recognition networks. Viterbi then finds the most likely path through the network. State sequence → subunit sequence → word sequence → sentence

Lexicon: Describes pronounciation of all words allowed; Language model: Describes all utterances allowed and their probabilities. Typically, Mel-Cepstrum, Delta-Cepstrum or other are used as features.

Disadvantages: HMM assumptions never totally met in practice: Conditional independence of states and observations; Training HMMs is not inherently discriminative but rather likelihood maximizing.

### 8.2.6 Deep Learning Speech Recognizer

**Datasets**: Split dataset into training and test/evaluation sets. **Word Error Rate**: Count substitutions (S), insertions (I), delections (D) and divide by number of words in ground truth. **Data Preprocessing** Scale input data to have zero mean and unit variance. **Initialization**: use random gaussian distributed weights. **Overfitting**: Memorizing the training set. Use dropout to mitigate. **Batch Size**: Compromise between faster and more optimal training. **Batch Normalization**: Scale the activations to have zero mean and unit variance (only during training).

Normal Depp Neural Networks (DNN) classify only input patterns of constant size (e.g. image), so other architectures are used.

**DNN-HMM**: Use DNN as feature extractor for HMM; **CNN**: Use CNN on spectogram; **LSTM vs. GRU**: both types of recurrent neural network (RNN) architectures designed to address the vanishing gradient problem in traditional RNNs. LSTMs have a more complex memory cell that consists of a cell state and three gates - input gate, forget gate, and output gate. GRUs have a simpler memory cell with two gates - reset gate and update gate.