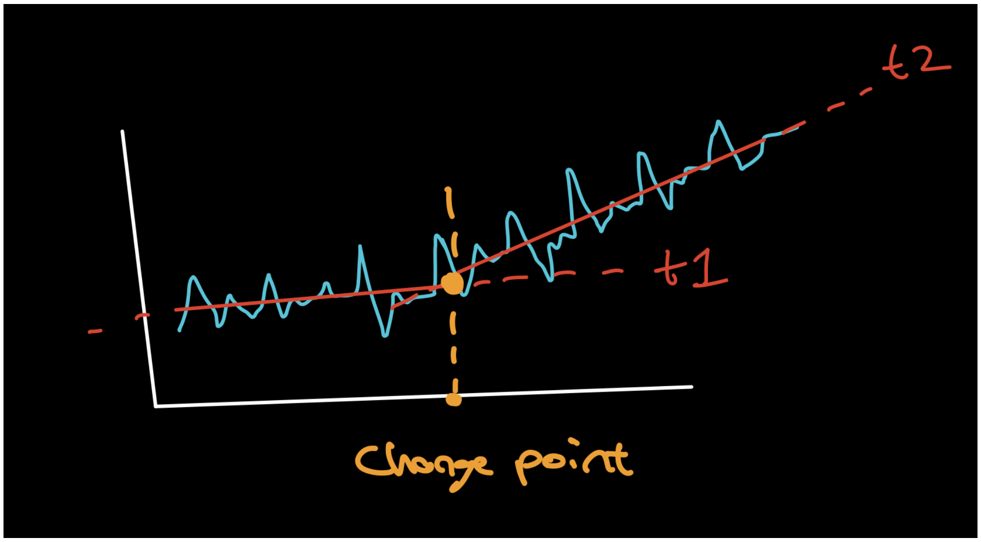
Time Series Analysis Lecture — 5

Change Points in Time Series Analysis



- **Concept:** A change point marks a significant shift in the trend or statistical properties of a time series, dividing it into segments with distinct characteristics.
- **Detection:**
  - **Sliding Window Method:** A simple approach using a fixed-size window to traverse the series, calculating a cost (e.g., slope change) for elements within the window. Peaks in this cost indicate potential change points.
  - **Libraries and Algorithms:** Advanced methods (e.g., ‘ruptures’ library) employ more sophisticated algorithms for detecting change points, offering improved accuracy and customization.
- **Utilization:**
  - **Analysis:** Change points help in understanding shifts in data trends, useful for retrospective analysis rather than forecasting future values.
  - **Modeling:** Identifying change points can refine model adjustments, especially when integrating these insights into dynamic models that can adapt to historical shifts.
- **Exogenous Variables and Change Points:**
- Incorporating known change points as exogenous variables in models like SARIMAX can potentially enhance forecast accuracy by accounting for structural breaks.
- **Practical Application:**
  - Demonstrating change point detection through code examples, utilizing both simple methods and libraries like ‘ruptures’, enriches understanding and application in real-world datasets.
- **Instructor Notes**:

    - Emphasize the distinction between detecting change points for analytical insights versus their limited direct utility in forecasting future observations.
    - Highlight the iterative nature of selecting parameters (window size, threshold) in simple methods and exploring library options for more complex scenarios.

Time Series Forecasting Using Linear Regression

- **Approach:** Applying linear regression to forecast time series data by engineering features that capture underlying patterns and seasonality.
- **Feature Engineering:**
    - **Weekday/Weekend:** Binary feature indicating weekends, capturing visitor variation.
    - **Lagged Features:** Past values (e.g., Lag_1, Lag_2, ...) to utilize historical data.
    - **Averaging:** Using past averages (e.g., last month, week, 2-week averages) to smooth out short-term fluctuations.
    - **Seasonality:** Average sales by day of the week to capture weekly seasonality.
- **Model Training:**
    - **Selected features:** Lag_1, last_month_avg_level, last_week_avg_level, last_2week_avg_level, sale_wrt_dow, holiday.
    - Split data into training and testing sets for model evaluation.
    - Train a Linear Regression model on the training set.
- **Performance Evaluation:**
    - Use metrics such as MAPE (Mean Absolute Percentage Error) to assess forecast accuracy.
    - Visualize actual vs. predicted values to qualitatively evaluate the model's performance.
- **Observations:**
    - The linear regression model can perform surprisingly well for forecasting, especially when predicting short-term future values.
    - Incorporating exogenous variables like holidays can significantly improve forecast accuracy.
- **Considerations for Improvement:**
    - Innovate with feature engineering to capture more complex patterns.
    - Apply feature selection techniques to refine the model.
    - Explore different regression models and hyperparameter tuning for optimization.
    - Consider stacking or cascading models for enhanced predictions.
- **Important Note:**
    - SARIMAX provided forecasts for a longer horizon, while the linear regression model focused on short-term (1-day ahead) forecasting using features like Lag_1.
    - For multi-day forecasts, a separate linear regression model is needed for each forecast horizon.
- **Instructor Notes:**

    - Highlight the importance of feature engineering in applying linear regression to time series forecasting.
    - Encourage experimentation with different features, models, and techniques to improve forecast accuracy.
    - Discuss the trade-offs between model simplicity and forecast horizon flexibility.

Facebook's Prophet:

Prophet is a forecasting tool by Facebook designed for time series data that is robust to missing data, shifts in trend, and outliers. It excels in handling datasets with strong seasonal effects.

- **Features:**
    - **Intuitive Parameters:** Easy-to-tune for quick adjustments.
    - **Handles Anomalies:** Efficiently manages missing data and outliers.
    - **Multiple Seasonalities:** Captures complex seasonal patterns through Fourier transforms.
    - **Decomposable Model:** Splits into trend, seasonality, holiday effects, plus an error term.
- **Model Components**:
    - **Trend g(t)):** Represents non-periodic changes.
    - **Seasonality s(t)):** Captures periodic changes (e.g., weekly, yearly).
    - **Holidays h(t)):** Models holiday effects with irregular schedules.
    - **Error (ε_t):** Accounts for unexplained changes.
- **Usage Highlights:**
    - Requires data with ‘ds’ (date-time) and ‘y’ (target variable) columns.
    - Easily incorporates holidays and external regressors for improved forecasts.
    - Allows flexibility adjustments via ‘changepoint_prior_scale’.
- **Benefits:**
    - **User-Friendly:** Minimal data preparation needed.
    - **Robust and Versatile:** Great for seasonal forecasts and handling data irregularities.
    - **Enhanced Forecasting:** Supports multiple seasonalities and holiday effects for accurate predictions.
    - **Interpretability:** Provides clear insights into the components of the forecast.

Prophet stands out for its simplicity and effectiveness, especially for time series with pronounced seasonal patterns, offering a strong baseline model with minimal feature engineering.