Time Series Forecasting: Comprehensive Summary

Mathematical Notes Collection

October 28, 2025

Contents

1 Introduction to Time Series Forecasting

Time series forecasting is the process of predicting future values of a time series based on its historical patterns. It combines statistical methods, machine learning techniques, and domain expertise to make accurate predictions about future trends, seasonality, and patterns.

1.1 Key Concepts

Definition 1.1 (Time Series)**.** *A time series is a sequence of data points collected at regular time intervals, typically denoted as* $\{X_t\}_{t=1}^{T}$ *where* t *represents time and* T *is the total number of observations.*

Definition 1.2 (Forecasting)**.** *Forecasting is the process of predicting future values* $\hat{X}_{T+h}$ *for horizon* h *based on historical data* $\{X_t\}_{t=1}^{T}$*.*

1.2 Components of Time Series

1. **Trend**: Long-term increase or decrease in the data
2. **Seasonality**: Regular patterns that repeat over fixed periods
3. **Cyclical**: Irregular patterns without fixed periodicity
4. **Irregular/Random**: Unpredictable noise and random fluctuations

1.3 Decomposition

Definition 1.3 (Additive Decomposition)**.**

$$X_t = T_t + S_t + C_t + \epsilon_t$$

where T_t *is trend,* S_t *is seasonal,* C_t *is cyclical, and* ϵ_t *is irregular component.*

Definition 1.4 (Multiplicative Decomposition)**.**

$$X_t = T_t \times S_t \times C_t \times \epsilon_t$$

2 Traditional Statistical Methods

2.1 Exponential Smoothing

2.1.1 Simple Exponential Smoothing

Definition 2.1 (Simple Exponential Smoothing)**.** *For a time series without trend or seasonality:*

$$\hat{X}_{t+1} = \alpha X_t + (1-\alpha)\hat{X}_t$$

where $\alpha \in [0, 1]$ *is the smoothing parameter.*

Algorithm 1 Simple Exponential Smoothing

```
Initialize X̂_1 = X_1
for t = 2 to T do
    X̂_t = αX_{t−1} + (1 − α)X̂_{t−1}
end for
Forecast: X̂_{T+h} = X̂_T for h ≥ 1
```

2.1.2 Holt's Method (Double Exponential Smoothing)

Definition 2.2 (Holt's Method)**.** *For time series with trend:*

$$\hat{X}_{t+1} = L_t + b_t$$

where:

$$L_t = \alpha X_t + (1-\alpha)(L_{t-1} + b_{t-1})$$
$$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}$$

2.1.3 Holt-Winters Method (Triple Exponential Smoothing)

Definition 2.3 (Holt-Winters Method)**.** *For time series with trend and seasonality:*

$$\hat{X}_{t+h} = (L_t + hb_t)S_{t+h-m}$$

where m is the seasonal period and:

$$L_t = \alpha\frac{X_t}{S_{t-m}} + (1-\alpha)(L_{t-1} + b_{t-1})$$
$$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}$$
$$S_t = \gamma\frac{X_t}{L_t} + (1-\gamma)S_{t-m}$$

2.2 ARIMA Models

2.2.1 Autoregressive (AR) Models

Definition 2.4 (AR(p) Model)**.** *An autoregressive model of order p:*

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \epsilon_t$$

where $\epsilon_t \sim N(0, \sigma^2)$ is white noise.

2.2.2 Moving Average (MA) Models

Definition 2.5 (MA(q) Model)**.** *A moving average model of order q:*

$$X_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \ldots + \theta_q\epsilon_{t-q}$$

2.2.3 ARIMA(p,d,q) Model

Definition 2.6 (ARIMA Model)**.** *An ARIMA(p,d,q) model for a time series $\{Y_t\}$:*

$$\phi(B)(1-B)^d Y_t = \theta(B)\epsilon_t$$

where B is the backshift operator, $(1-B)^d$ represents differencing d times, and:

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$$
$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q$$

2.2.4 Seasonal ARIMA (SARIMA)

Definition 2.7 (SARIMA(p,d,q)(P,D,Q)$_s$ Model)**.**

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D Y_t = \theta(B)\Theta(B^s)\epsilon_t$$

where s is the seasonal period and $\Phi(B^s)$, $\Theta(B^s)$ are seasonal polynomials.

2.3 State Space Models

Definition 2.8 (State Space Model)**.** *A state space model consists of:*

$$\textit{State equation: } \mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\mathbf{w}_t \tag{1}$$

$$\textit{Observation equation: } \mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \tag{2}$$

where $\mathbf{w}_t$ *and* $\mathbf{v}_t$ *are noise processes.*

2.3.1 Kalman Filter

Algorithm 2 Kalman Filter

Initialize $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$
for $t = 1$ to T **do**
  Prediction:
  $\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1}$
  $\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T$
  Update:
  $\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R})^{-1}$
  $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1})$
  $\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}$
end for

3 Machine Learning Methods

3.1 Linear Models

3.1.1 Linear Regression

Definition 3.1 (Linear Regression for Time Series)**.**

$$\hat{X}_{t+h} = \beta_0 + \sum_{i=1}^{p} \beta_i X_{t-i+1} + \sum_{j=1}^{q} \gamma_j f_j(t)$$

where $f_j(t)$ *are time-based features (trend, seasonality, etc.).*

3.1.2 Ridge and Lasso Regression

Definition 3.2 (Ridge Regression)**.**

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{t=1}^{T} (X_t - \mathbf{x}_t^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

Definition 3.3 (Lasso Regression)**.**

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{t=1}^{T} (X_t - \mathbf{x}_t^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

3.2 Tree-Based Methods

3.2.1 Random Forest

Algorithm 3 Random Forest for Time Series

1: Create bootstrap samples from training data
2: **for** each bootstrap sample **do**
3: Train decision tree with random feature selection
4: Use lagged values and time features as inputs
5: **end for**
6: Predict: Average predictions from all trees

3.2.2 Gradient Boosting

Definition 3.4 (Gradient Boosting)**.**

$$\hat{F}_m(\mathbf{x}) = \hat{F}_{m-1}(\mathbf{x}) + \gamma_m h_m(\mathbf{x})$$

where h_m is the m-th weak learner and γ_m is the step size.

3.3 Neural Networks

3.3.1 Feedforward Neural Networks

Definition 3.5 (MLP for Time Series)**.**

$$\hat{X}_{t+h} = f(\mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x}_t + \mathbf{b}_1) + \mathbf{b}_2)$$

where $\mathbf{x}_t = [X_{t-1}, X_{t-2}, \ldots, X_{t-p}]^T$ and σ is activation function.

3.3.2 Recurrent Neural Networks (RNN)

Definition 3.6 (RNN)**.**

$$\mathbf{h}_t = \sigma(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{b}_h)$$
$$\hat{X}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y$$

3.3.3 Long Short-Term Memory (LSTM)

Definition 3.7 (LSTM Cell)**.**

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{3}$$
$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{4}$$
$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{5}$$
$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{C}}_t \tag{6}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{7}$$
$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{C}_t) \tag{8}$$

3.3.4 Gated Recurrent Unit (GRU)

Definition 3.8 (GRU Cell)**.**

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{9}$$
$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{10}$$
$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W} \cdot [\mathbf{r}_t * \mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{11}$$
$$\mathbf{h}_t = (1 - \mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t \tag{12}$$

3.4 Transformer-Based Models

3.4.1 Time Series Transformer

Definition 3.9 (Time Series Transformer)**.**

$$\textit{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \textit{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

where $\mathbf{Q}$*,* $\mathbf{K}$*,* $\mathbf{V}$ *are derived from time series embeddings.*

3.4.2 Informer

Definition 3.10 (Informer Architecture)**.** *The Informer model uses:*

- *ProbSparse self-attention mechanism*
- *Self-attention distilling operation*
- *Generative style decoder*

3.5 Deep Learning Architectures

3.5.1 CNN-LSTM

Definition 3.11 (CNN-LSTM)**.**

$$\mathbf{c}_t = \textit{CNN}(\mathbf{x}_{t-w:t})$$
$$\mathbf{h}_t = \textit{LSTM}(\mathbf{c}_t, \mathbf{h}_{t-1})$$
$$\hat{X}_{t+h} = \mathbf{W}\mathbf{h}_t + \mathbf{b}$$

3.5.2 Seq2Seq Models

Definition 3.12 (Sequence-to-Sequence)**.**

$$\mathbf{h}_t = \textit{Encoder}(\mathbf{x}_t, \mathbf{h}_{t-1})$$
$$\mathbf{s}_t = \textit{Decoder}(\mathbf{s}_{t-1}, \mathbf{h}_T)$$
$$\hat{X}_{t+h} = \textit{Output}(\mathbf{s}_t)$$

4 Feature Engineering

4.1 Time-Based Features

1. **Temporal Features**: Hour, day, week, month, year
2. **Cyclical Encoding**: $\sin(2\pi t/p)$, $\cos(2\pi t/p)$
3. **Lagged Features**: $X_{t-1}, X_{t-2}, \ldots, X_{t-p}$
4. **Rolling Statistics**: Mean, std, min, max over windows
5. **Difference Features**: $\Delta X_t = X_t - X_{t-1}$

4.2 Statistical Features

Definition 4.1 (Rolling Window Statistics)**.** *For window size w:*

$$Mean_t = \frac{1}{w}\sum_{i=0}^{w-1} X_{t-i}$$

$$Std_t = \sqrt{\frac{1}{w-1}\sum_{i=0}^{w-1}(X_{t-i} - Mean_t)^2}$$

4.3 Domain-Specific Features

1. **Weather Data**: Temperature, humidity, pressure
2. **Economic Indicators**: GDP, inflation, interest rates
3. **Calendar Events**: Holidays, special events
4. **External Factors**: Market conditions, news sentiment

5 Model Selection and Validation

5.1 Time Series Cross-Validation

Algorithm 4 Time Series Cross-Validation

```
Split data into k folds chronologically
for i = 1 to k do
    Train on folds 1 to i − 1
    Validate on fold i
    Compute validation error
end for
Average validation errors across folds
```

5.2 Walk-Forward Analysis

Definition 5.1 (Walk-Forward Analysis)**.** *For each time point t:*

1. *Train model on data up to time t*
2. *Predict value at time $t + h$*
3. *Move to time $t + 1$ and repeat*

5.3 Model Evaluation Metrics

Definition 5.2 (Mean Absolute Error (MAE))**.**

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |X_i - \hat{X}_i|$$

Definition 5.3 (Root Mean Square Error (RMSE))**.**

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2}$$

Definition 5.4 (Mean Absolute Percentage Error (MAPE))**.**

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{X_i - \hat{X}_i}{X_i} \right|$$

Definition 5.5 (Symmetric MAPE (sMAPE))**.**

$$sMAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|X_i - \hat{X}_i|}{(|X_i| + |\hat{X}_i|)/2}$$

6 Advanced Topics

6.1 Multivariate Time Series

Definition 6.1 (Vector Autoregression (VAR))**.**

$$\mathbf{X}_t = \mathbf{c} + \mathbf{A}_1 \mathbf{X}_{t-1} + \mathbf{A}_2 \mathbf{X}_{t-2} + \ldots + \mathbf{A}_p \mathbf{X}_{t-p} + \boldsymbol{\epsilon}_t$$

where $\mathbf{X}_t$ *is a vector of time series.*

6.2 Nonlinear Models

6.2.1 Threshold Autoregression (TAR)

Definition 6.2 (TAR Model)**.**

$$X_t = \begin{cases} \phi_1^{(1)} + \phi_1^{(1)} X_{t-1} + \ldots + \phi_p^{(1)} X_{t-p} + \epsilon_t^{(1)} & \text{if } X_{t-d} \leq r \\ \phi_0^{(2)} + \phi_1^{(2)} X_{t-1} + \ldots + \phi_p^{(2)} X_{t-p} + \epsilon_t^{(2)} & \text{if } X_{t-d} > r \end{cases}$$

6.3 Ensemble Methods

Definition 6.3 (Ensemble Forecast)**.**

$$\hat{X}_t = \sum_{i=1}^{M} w_i \hat{X}_{t,i}$$

where $\sum_{i=1}^{M} w_i = 1$ *and* $\hat{X}_{t,i}$ *are individual forecasts.*

6.4 Probabilistic Forecasting

Definition 6.4 (Quantile Regression)**.** *For quantile* τ*:*

$$\hat{Q}_\tau(X_{t+h}) = \arg\min_{\theta} \sum_{i=1}^{n} \rho_\tau(X_i - \theta)$$

where $\rho_\tau(u) = u(\tau - \mathbf{1}_{u<0})$*.*

7 Applications

7.1 Economic Forecasting

1. **GDP Growth**: Economic output prediction
2. **Inflation Rates**: Price level forecasting
3. **Unemployment**: Labor market predictions
4. **Stock Prices**: Financial market forecasting

7.2 Demand Forecasting

1. **Retail Sales**: Product demand prediction
2. **Energy Consumption**: Power grid planning
3. **Transportation**: Traffic flow prediction
4. **Healthcare**: Patient volume forecasting

7.3 Weather and Climate

1. **Temperature**: Weather prediction
2. **Precipitation**: Rainfall forecasting
3. **Climate Change**: Long-term climate modeling
4. **Renewable Energy**: Solar/wind power prediction

7.4 Technology and IoT

1. **Server Load**: IT infrastructure planning
2. **Sensor Data**: IoT device monitoring
3. **Network Traffic**: Bandwidth forecasting
4. **User Behavior**: Web analytics prediction

8 Challenges and Considerations

8.1 Data Quality Issues

1. **Missing Values**: Handling gaps in time series
2. **Outliers**: Detecting and treating anomalies
3. **Non-stationarity**: Addressing trend and seasonality
4. **Data Sparsity**: Working with limited historical data

8.2 Model Complexity

1. **Overfitting**: Balancing model complexity and generalization
2. **Hyperparameter Tuning**: Optimizing model parameters
3. **Computational Cost**: Managing training time and resources
4. **Interpretability**: Understanding model decisions

8.3 External Factors

1. **Regime Changes**: Handling structural breaks
2. **External Shocks**: Accounting for unexpected events
3. **Correlation Changes**: Managing time-varying relationships
4. **Nonlinearity**: Capturing complex patterns

9 Software and Tools

9.1 Statistical Software

1. **R**: forecast, fable, tsibble packages
2. **Python**: statsmodels, pmdarima, sktime
3. **MATLAB**: Econometrics Toolbox
4. **SAS**: SAS/ETS, SAS Forecast Server

9.2 Machine Learning Libraries

1. **Python**: scikit-learn, TensorFlow, PyTorch
2. **R**: caret, randomForest, xgboost
3. **Julia**: Flux.jl, MLJ.jl
4. **Scala**: Spark MLlib

9.3 Specialized Tools

1. **Prophet**: Facebook's forecasting tool
2. **NeuralProphet**: Neural network-based Prophet
3. **Darts**: Python library for time series
4. **GluonTS**: Probabilistic time series modeling

10 Key Theorems and Results

Theorem 10.1 (Wold's Decomposition)**.** *Any covariance-stationary time series can be decomposed into a deterministic component and a purely non-deterministic component.*

Theorem 10.2 (Granger Causality)**.** *A time series X Granger-causes Y if past values of X contain information that helps predict Y beyond what is contained in past values of Y alone.*

Proposition 10.3 (Stationarity Requirement)**.** *For ARIMA models to be valid, the time series must be stationary or made stationary through differencing.*

Theorem 10.4 (Optimal Forecast)**.** *The optimal forecast minimizes the expected squared prediction error under certain regularity conditions.*

11 Future Directions

11.1 Emerging Methods

1. **Graph Neural Networks**: Modeling complex dependencies
2. **Attention Mechanisms**: Focusing on relevant time periods
3. **Generative Models**: Probabilistic forecasting
4. **Transfer Learning**: Leveraging related time series

11.2 Real-Time Forecasting

1. **Streaming Algorithms**: Online learning approaches
2. **Edge Computing**: Distributed forecasting systems
3. **Adaptive Models**: Self-updating algorithms
4. **Low-Latency**: Fast prediction systems

11.3 Interpretable AI

1. **Explainable Models**: Understanding predictions
2. **Causal Inference**: Identifying causal relationships
3. **Counterfactual Analysis**: What-if scenarios
4. **Uncertainty Quantification**: Confidence intervals

12 Conclusion

Time series forecasting is a critical field that combines statistical rigor with machine learning innovation. The choice between traditional statistical methods and modern ML approaches depends on:

- **Data characteristics**: Size, quality, and complexity
- **Forecasting horizon**: Short-term vs. long-term predictions
- **Interpretability requirements**: Need for explainable models
- **Computational constraints**: Available resources and time

Traditional methods like ARIMA and exponential smoothing remain valuable for their interpretability and statistical foundations, while machine learning approaches excel at capturing complex patterns and handling high-dimensional data.

The future of time series forecasting lies in:

- Hybrid approaches combining statistical and ML methods
- Real-time adaptive systems
- Probabilistic forecasting with uncertainty quantification
- Integration of external data sources and domain knowledge

As data availability increases and computational power grows, the field continues to evolve toward more sophisticated, accurate, and interpretable forecasting systems that can handle the complexity of real-world time series data.