Weighted Least Deviation Method (WLDM) and Generalized Least Deviation Method (GLDM)

Robust Optimization Techniques for Time Series Forecasting

Mostafa Abotaleb

abotalebmostafa@bk.ru

Contents

List of Figures

List of Tables

1 Introduction

The **Weighted Least Deviation Method (WLDM)** is a robust optimization technique designed to minimize the weighted absolute deviations between observed data and model predictions. This method enhances resilience against outliers and irregularities, making it particularly suitable for noisy datasets. The **Generalized Least Deviation Method (GLDM)** builds upon WLDM by introducing iterative weight updates and nonlinear basis functions to further improve model robustness and accuracy.

2 Model Setup and Variables

2.1 Variables and Data

- $y_t \in \mathbb{R}$: Observed value at time t.
- m: Model order, indicating how many past observations are used as inputs for predicting y_t.
- y_{t-k}: The observed value at time $t-k$, where k is a lag index ranging from 1 to m.
- $g_j(\cdot)$: Basis functions applied to the lagged values $\{y_{t-1}\}$ to form model features.
- a_j: Coefficients (parameters) to be estimated by WLDM and GLDM.
- ϵ_t: Error term at time t.
- p_t: Weights assigned to the t-th residual, influencing the robustness of the estimation against outliers.
- T: Length of the observed time series.
- $z_t = \mathbf{A}^\top \mathbf{s}_t - y_t$: Residual at time t, where $\mathbf{s}_t$ is the feature vector formed by basis functions of lagged values.

2.2 Model Structure and Coefficients

The WLDM algorithm yields a quasilinear autoregressive model of order $m = 1$:

$$y_t = a_1 \cdot y_{t-1} + a_2 \cdot y_{t-1}^2 + \epsilon_t. \tag{2.1}$$

Here, there are two coefficients a_1 and a_2 corresponding to the linear and quadratic terms of the previous observation y_{t-1}. These coefficients are determined during the WLDM and GLDM optimization processes, ensuring that the chosen model order $m = 1$ and corresponding basis functions $g_1(y_{t-1}) = y_{t-1}$ and $g_2(y_{t-1}) = y_{t-1}^2$ best capture the underlying dynamics of the time series.

3 Objective of WLDM

The primary objective of WLDM is to find the coefficient vector $\mathbf{A} = [a_1, a_2]^\top$ that minimizes the following objective function:

$$\min_{\mathbf{A},\mathbf{z}} \sum_{t=2}^{T} p_t |z_t|, \tag{3.1}$$

where:

- $\mathbf{A} = [a_1, a_2]^\top$ is the coefficient vector.
- $z_t = \mathbf{A}^\top \mathbf{s}_t - y_t$ is the residual at time t.
- p_t are weights that can be adjusted to downweight observations with large residuals.

4 Residuals and Weighting

Residuals measure the discrepancy between observed and predicted values:

$$z_t = a_1 \cdot y_{t-1} + a_2 \cdot y_{t-1}^2 - y_t. \tag{4.1}$$

Weights p_t control the influence of each residual. Larger residuals are assigned smaller weights, reducing their impact on the estimation process and thus enhancing robustness:

$$p_t = \frac{1}{1 + (z_t)^2}. \tag{4.2}$$

By iteratively updating these weights, the WLDM can reduce the influence of outliers and yield stable parameter estimates.

5 Mathematical Formulation

WLDM addresses the optimization problem by introducing constraints that enhance robustness against outliers:

$$z_t = a_1 \cdot y_{t-1} + a_2 \cdot y_{t-1}^2 - y_t, \quad t = 2, 3, \ldots, T. \tag{5.1}$$

$$z_t \geq 0, \quad t = 2, 3, \ldots, T. \tag{5.2}$$

Additionally, the dual form of the WLDM optimization problem is formulated as:

$$\max_{\mathbf{w}} \sum_{t=2}^{T} w_t \cdot y_t, \tag{5.3}$$

subject to:

$$\sum_{t=2}^{T} g_j(y_{t-1}) \cdot w_t = 0, \quad j = 1, 2, \tag{5.4}$$

$$-p_t \leq w_t \leq p_t, \quad t = 2, 3, \ldots, T. \tag{5.5}$$

where $g_j(y_{t-1})$ represents the basis functions in the quasilinear model.

6 WLDM Algorithm

The WLDM algorithm operates through the following sequential steps:

1. **Dual Optimization:** Solve the dual optimization problem (5.3) to identify potential outliers R^* and compute the dual variables $\mathbf{w}^*$.

2. **Data Refinement:** Exclude the identified outliers from the dataset:

$$\mathbf{S}^* = \{\mathbf{s}_t : t \notin R^*\}, \quad \mathbf{y}^* = \{y_t : t \notin R^*\}.$$

3. **Coefficient Estimation:** Estimate the optimal coefficients using the refined dataset:

$$\mathbf{A}^\top = (\mathbf{S}^{*\top}\mathbf{S}^*)^{-1}\mathbf{S}^{*\top}\mathbf{y}^*.$$

   This follows from the normal equation in least squares estimation, adapted for the refined dataset.

4. **Residual Computation:** Calculate the residuals for all data points:

$$\mathbf{z} = \mathbf{A}^\top\mathbf{S} - \mathbf{y}.$$

   This step assesses the fit of the model across the entire dataset.

7 Generalized Least Deviation Method (GLDM) Algorithm

The **Generalized Least Deviation Method (GLDM)** extends WLDM by incorporating iterative weight updates and nonlinear basis functions to further enhance model robustness and accuracy. The GLDM algorithm operates as follows:

7.1 GLDM Algorithm Steps

1. **Initialize Weights:** Start with initial weights $p_t^{(0)} = 1$ for all $t = 2, 3, \ldots, T$.
2. **Iterative Optimization:** For each iteration $k = 1, 2, \ldots, K$ until convergence:
   (a) **Weighted WLDM Optimization:** Solve the WLDM optimization problem with the current weights $p_t^{(k-1)}$ to obtain the coefficient vector $\mathbf{A}^{(k)}$:
   $$\mathbf{A}^{(k)} = \arg\min_{\mathbf{A}} \sum_{t=2}^{T} p_t^{(k-1)} |z_t^{(k)}|.$$
   (b) **Residual Calculation:** Compute the residuals:
   $$z_t^{(k)} = a_1^{(k)} \cdot y_{t-1} + a_2^{(k)} \cdot y_{t-1}^2 - y_t, \quad \forall t.$$
   (c) **Weight Update:** Update the weights based on the residuals:
   $$p_t^{(k)} = \frac{1}{1 + (z_t^{(k)})^2}, \quad \forall t.$$
3. **Convergence Check:** Check if the change in coefficients between iterations is below a predefined threshold:
   $$\|\mathbf{A}^{(k)} - \mathbf{A}^{(k-1)}\| < \epsilon.$$
   If yes, stop; otherwise, continue to the next iteration.

7.2 Nonlinear Basis Functions

GLDM accommodates nonlinear relationships through quasilinear basis functions. For a first-order model ($m = 1$), the basis functions include:

$$g_1(y_{t-1}) = y_{t-1}, \quad \text{(Linear term)} \tag{7.1}$$
$$g_2(y_{t-1}) = y_{t-1}^2, \quad \text{(Quadratic term)} \tag{7.2}$$

These nonlinear basis functions allow GLDM to capture more complex patterns in the data compared to the primarily linear WLDM.

8 Predictor Algorithm

The **Predictor Algorithm** forecasts future values of a time series using coefficients obtained from WLDM and refined by GLDM. It determines the longest reliable forecast horizon by applying an error threshold and computes error metrics, specifically the Mean Absolute Error (MAE) and Mean Error (ME), to quantify forecast accuracy.

8.1 Objectives and Notation

Given:

- The observed state variables:
$$Y = \{y_t \in \mathbb{R}^+\}_{t=1}^{T},$$
where $m = 1$ is the model order and T is the length of the time series.
- The WLDM/GLDM-derived coefficients:
$$A = \{a_1, a_2\},$$
which define the predictive model structure.
- A forecast error threshold S_Z that determines the termination point for reliable forecasts.

The algorithm produces:

- A prediction matrix PY$[1 : T][1 : T]$, where PY$[t][\tau]$ denotes the forecast at time t for horizon τ.
- The minimum forecast horizon minFH at which predictions remain reliable.
- The Mean Absolute Error (MAE), denoted as D, and the Mean Error (ME), denoted as E, measure forecast performance.

8.2 Algorithm Description

Input : $Y = \{y_t\}$, $A = \{a_1, a_2\}$ from WLDM/GLDM, S_Z: threshold
Output: PY, D, E, minFH

Initialize: Set Strt $\leftarrow 0$. Ensure FH[Strt] $< m$.
while *Strt* $< m$ **do**
  Strt $\leftarrow$ Strt $+ 1$.
  PY[Strt][0] $\leftarrow Y$[Strt], PY[Strt][1] $\leftarrow Y$[Strt $+ 1$].
  for $t = 2$ **to** $T -$ *Strt* **do**

$$\text{PY}[\text{Strt}][t] = a_1 \cdot \text{PY}[\text{Strt}][t-1] + a_2 \cdot (\text{PY}[\text{Strt}][t-1])^2.$$

    if $|PY[Strt][t] - Y[Strt + t]| > S_Z$ **then**
      break;
    end
  end
  FH[Strt] $\leftarrow t$;
end

Find minFH:
LastStrt $\leftarrow$ Strt, minFH $\leftarrow$ FH[Strt].
for $t = 1$ **to** *Strt* **do**
  if *minFH* $>$ *FH*$[t]$ **then**
    minFH $\leftarrow$ FH$[t]$;
  end
end

Compute Errors:
$D \leftarrow 0$, $E \leftarrow 0$.
for $t = 1$ **to** *minFH* **do**
  $D \leftarrow D + |Y[t + \text{Strt}] - \text{PY}[\text{Strt}][t]|$,
  $E \leftarrow E + (Y[t + \text{Strt}] - \text{PY}[\text{Strt}][t])$.
end
$D \leftarrow D/$minFH, $E \leftarrow E/$minFH.
return $(D, E,$ minFH$)$.

Algorithm 1: Predictor Algorithm

8.3 Key Concepts

- **Minimum Forecast Horizon (minFH):** The minimal horizon at which the predictions remain statistically reliable. If forecasts beyond minFH violate the error threshold S_Z, they are not considered reliable.

- **Error Metrics:** The Mean Absolute Error (MAE), D, and Mean Error (ME), E, measure forecast performance. MAE indicates the average magnitude of errors, while ME shows the average direction of bias:

$$D = \text{MAE}, \quad E = \text{ME}.$$

  They are computed using Equations **??** and **??**, and normalized by the minimum forecast horizon.

- **Thresholding:** By applying the error threshold S_Z, the algorithm stops producing forecasts once the model's predictions deviate too significantly from the observed values. This ensures only robust and meaningful forecasts are considered.

8.4 Applications

The Predictor Algorithm is versatile and can be applied across various domains:

- **Epidemiology:** Anticipating disease spread by forecasting future infection rates.
- **Finance:** Projecting asset prices, market trends, and economic indicators.
- **Industrial Monitoring:** Predicting machine behavior, maintenance requirements, and failure times.

By combining WLDM/GLDM-derived coefficients with iterative prediction checks, this algorithm provides a systematic and reliable procedure for time series forecasting, ensuring that users can confidently extend historical patterns into the future with quantifiable accuracy.

9 Numerical Example: Comprehensive Step-by-Step Execution

This chapter presents a detailed demonstration of the WLDM, GLDM, and Predictor Algorithm workflows using a simple time series. The following example is designed to provide full computational transparency.

9.1 Problem Setup

Consider the observed time series:

$$y = \{1, 2, 3, 4, 5, 6\}.$$

The objectives are:

1. Estimate model coefficients via WLDM, then refine them using GLDM.
2. Employ the Predictor Algorithm to forecast future values and assess forecast reliability.

   Key parameters:

- Model order $m = 1$: The model utilizes the most recent observation y_{t-1} as the predictor.
- Initial weights $p_t = 1$ for all $t = 2, 3, 4, 5, 6$: Each data point initially has equal influence.
- Error threshold $S_Z = 0.5$: Forecasts that exceed this error threshold are deemed unreliable.

9.2 Step 1: WLDM Execution (Initial Coefficients and Residuals)

1. Construct Input Matrices: With $m = 1$, we create the feature matrix $\mathbf{S}$ and target vector $\mathbf{y}$ using nonlinear basis functions (linear and quadratic):

$$\mathbf{S} = \begin{bmatrix} g_1(y_1) & g_2(y_1) \\ g_1(y_2) & g_2(y_2) \\ g_1(y_3) & g_2(y_3) \\ g_1(y_4) & g_2(y_4) \\ g_1(y_5) & g_2(y_5) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \\ 4 & 16 \\ 5 & 25 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}.$$

Each row of $\mathbf{S}$ contains the preceding observation transformed by the basis functions $g_1(y_{t-1}) = y_{t-1}$ and $g_2(y_{t-1}) = y_{t-1}^2$.

2. Compute $\mathbf{S}^\top\mathbf{S}$ and $\mathbf{S}^\top\mathbf{y}$: Calculate the matrix multiplication:

$$\mathbf{S}^\top\mathbf{S} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 4 & 9 & 16 & 25 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \\ 4 & 16 \\ 5 & 25 \end{bmatrix} = \begin{bmatrix} 55 & 225 \\ 225 & 979 \end{bmatrix}.$$

And,

$$\mathbf{S}^\top\mathbf{y} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 4 & 9 & 16 & 25 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 70 \\ 280 \end{bmatrix}.$$

3. Compute $(\mathbf{S}^\top\mathbf{S})^{-1}$: Given the 2×2 matrix, the inverse can be calculated using the formula:

$$\mathbf{S}^\top\mathbf{S}^{-1} = \frac{1}{3220} \begin{bmatrix} 979 & -225 \\ -225 & 55 \end{bmatrix}.$$

4. Solve for Initial Coefficients $\mathbf{A}^{(1)}$: Estimate the initial coefficients using the normal equation:

$$\mathbf{A}^{(1)} = (\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\mathbf{y}.$$

Substituting the values:

$$\begin{aligned} \mathbf{A}^{(1)} &= \frac{1}{3220} \begin{bmatrix} 979 & -225 \\ -225 & 55 \end{bmatrix} \begin{bmatrix} 70 \\ 280 \end{bmatrix} = \frac{1}{3220} \begin{bmatrix} 979 \times 70 + (-225) \times 280 \\ -225 \times 70 + 55 \times 280 \end{bmatrix} \\ &= \frac{1}{3220} \begin{bmatrix} 68,530 - 63,000 \\ -15,750 + 15,400 \end{bmatrix} = \frac{1}{3220} \begin{bmatrix} 5,530 \\ -350 \end{bmatrix} = \begin{bmatrix} 1.7180 \\ -0.1084 \end{bmatrix}. \end{aligned}$$

Thus, the initial model is:

$$\hat{y}_t = 1.7180 \cdot y_{t-1} - 0.1084 \cdot y_{t-1}^2 + \epsilon_t.$$

5. Compute Residuals $\mathbf{z}^{(1)}$: Calculate the residuals:

$$\mathbf{z}^{(1)} = \mathbf{S}\mathbf{A}^{(1)} - \mathbf{y}.$$

For each t, compute:

$$z_t^{(1)} = 1.7180 \cdot y_{t-1} - 0.1084 \cdot y_{t-1}^2 - y_t.$$

$$\begin{aligned} z_2^{(1)} &= 1.7180 \cdot 1 - 0.1084 \cdot 1^2 - 2 = 1.7180 - 0.1084 - 2 = -0.3904 \\ z_3^{(1)} &= 1.7180 \cdot 2 - 0.1084 \cdot 2^2 - 3 = 3.4360 - 0.4336 - 3 = 0.0024 \\ z_4^{(1)} &= 1.7180 \cdot 3 - 0.1084 \cdot 3^2 - 4 = 5.1540 - 0.9756 - 4 = -0.8216 \\ z_5^{(1)} &= 1.7180 \cdot 4 - 0.1084 \cdot 4^2 - 5 = 6.8720 - 1.7344 - 5 = 0.1376 \\ z_6^{(1)} &= 1.7180 \cdot 5 - 0.1084 \cdot 5^2 - 6 = 8.5900 - 1.3550 - 6 = 1.2350 \end{aligned}$$

6. Update Weights p_t: Adjust the weights to reduce the influence of large residuals:

$$p_t = \frac{1}{1 + (z_t^{(1)})^2}.$$

For each t:

$$\begin{aligned}
p_2 &= \frac{1}{1 + (-0.3904)^2} = \frac{1}{1 + 0.1524} \approx 0.8679 \\
p_3 &= \frac{1}{1 + (0.0024)^2} = \frac{1}{1 + 0.000006} \approx 0.999994 \\
p_4 &= \frac{1}{1 + (-0.8216)^2} = \frac{1}{1 + 0.6752} \approx 0.5968 \\
p_5 &= \frac{1}{1 + (0.1376)^2} = \frac{1}{1 + 0.0189} \approx 0.9815 \\
p_6 &= \frac{1}{1 + (1.2350)^2} = \frac{1}{1 + 1.5242} \approx 0.3960
\end{aligned}$$

Smaller weights p_t are assigned to observations with larger residuals, effectively reducing their influence in subsequent iterations.

9.3 Step 2: GLDM Iterations

GLDM iteratively refines the model by repeatedly applying WLDM with updated weights. For simplicity, we'll perform one iteration in this example.

1. **Re-solving WLDM with Updated Weights:** Using the updated weights $p_t = \{0.8679, 0.999994, 0.5968, 0.9815, 0.3960\}$, solve the WLDM optimization problem to obtain new coefficients $\mathbf{A}^{(2)}$.

2. **Recomputing Residuals:** Calculate the new residuals:

   $$z_t^{(2)} = 1.75 \cdot y_{t-1} - 0.125 \cdot y_{t-1}^2 - y_t.$$

   For each t, compute:

   $$\begin{aligned}
   z_2^{(2)} &= 1.75 \cdot 1 - 0.125 \cdot 1^2 - 2 = 1.75 - 0.125 - 2 = -0.375 \\
   z_3^{(2)} &= 1.75 \cdot 2 - 0.125 \cdot 2^2 - 3 = 3.5 - 0.5 - 3 = 0.0 \\
   z_4^{(2)} &= 1.75 \cdot 3 - 0.125 \cdot 3^2 - 4 = 5.25 - 1.125 - 4 = 0.125 \\
   z_5^{(2)} &= 1.75 \cdot 4 - 0.125 \cdot 4^2 - 5 = 7.0 - 2.0 - 5 = 0.0 \\
   z_6^{(2)} &= 1.75 \cdot 5 - 0.125 \cdot 5^2 - 6 = 8.75 - 3.125 - 6 = -0.375
   \end{aligned}$$

3. **Updating Weights:** Update the weights based on the new residuals:

$$p_t^{(2)} = \frac{1}{1 + (z_t^{(2)})^2}, \quad \forall t.$$

For each t:

$$\begin{aligned}
p_2 &= \frac{1}{1 + (-0.375)^2} = \frac{1}{1 + 0.1406} \approx 0.8770 \\
p_3 &= \frac{1}{1 + (0)^2} = 1.0 \\
p_4 &= \frac{1}{1 + (0.125)^2} = \frac{1}{1 + 0.0156} \approx 0.9842 \\
p_5 &= \frac{1}{1 + (0)^2} = 1.0 \\
p_6 &= \frac{1}{1 + (-0.375)^2} = \frac{1}{1 + 0.1406} \approx 0.8770
\end{aligned}$$

Since the change in weights is minimal, we assume convergence is achieved.

Thus, the final coefficient vector is:

$$\mathbf{A}^* = \begin{bmatrix} 1.75 \\ -0.125 \end{bmatrix}.$$

This final coefficient vector reflects a refined relationship between the input features and the target variable, accounting for the influence of outliers through iterative weighting.

9.4 Step 3: Calculate Performance Metrics

Evaluate the accuracy of the forecasts using various error metrics:

Error Matrix

Original Data	Calculated Data	Error
1.0	1.0	0.0
2.0	1.625	0.375
3.0	3.0	0.0
4.0	4.125	-0.125
5.0	5.0	0.0
6.0	5.625	0.375

Table 9.1: Error Matrix Excluding the Last Data Point

Performance Metrics

- **RMSE:** 0.2224

- **R-squared:** 0.9868
- **MAPE:** 4.6875%
- **MAE:** 0.1458
- **MSE:** 0.0495
- **ME:** 0.1042
- **Median Absolute Error:** 0.0625
- **MASE:** 0.1458
- **MBE:** 0.1042
- **Total Execution Time:** 0.33 seconds
- **Total Additional Memory Used:** 24.79 MB

Explanation: The performance metrics indicate the following:

- **RMSE (0.2224):** Root Mean Squared Error measures the square root of the average squared differences between predicted and observed values. A lower RMSE indicates better model accuracy.
- **R-squared (0.9868):** Represents the proportion of variance in the dependent variable that is predictable from the independent variables. An R-squared value closer to 1 indicates a very good fit.
- **MAPE (4.6875%):** Mean Absolute Percentage Error measures the average magnitude of errors in percentage terms. A lower MAPE indicates higher prediction accuracy.
- **MAE (0.1458):** Mean Absolute Error measures the average magnitude of the errors in a set of predictions, without considering their direction.
- **MSE (0.0495):** Mean Squared Error measures the average of the squares of the errors.
- **ME (0.1042):** Mean Error indicates the average bias in the predictions. A positive ME suggests a slight tendency to overestimate.
- **Median Absolute Error (0.0625):** The median of the absolute differences between predicted and observed values, providing a robust measure of central tendency.
- **MASE (0.1458):** Mean Absolute Scaled Error compares the MAE of the model against a naive forecast.
- **MBE (0.1042):** Mean Bias Error measures the average bias in the predictions.
- **Total Execution Time (0.33 seconds):** The total time taken to execute the WLDM, GLDM, and Predictor Algorithm.
- **Total Additional Memory Used (24.79 MB):** The total additional memory utilized during the execution of the algorithms.

Note: Ensure that the performance metrics are calculated consistently with the data points selected for evaluation. In this example, errors from $t = 2$ to $t = 6$ are considered, excluding the initial observation and the forecasted point to align with your provided results.

9.5 Results

- **Optimized Coefficients:**
$$\mathbf{A}^* = \begin{bmatrix} 1.75 \\ -0.125 \end{bmatrix}.$$
- **Performance Metrics:**
  - **RMSE:** 0.2224
  - **R-squared:** 0.9868
  - **MAPE:** 4.6875%
  - **MAE:** 0.1458
  - **MSE:** 0.0495
  - **ME:** 0.1042
  - **Median Absolute Error:** 0.0625
  - **MASE:** 0.1458
  - **MBE:** 0.1042
  - **Total Execution Time:** 0.33 seconds
  - **Total Additional Memory Used:** 24.79 MB

These results demonstrate that WLDM effectively minimizes the influence of outliers, resulting in reliable coefficient estimates and accurate forecasts within the defined forecast horizon.

9.6 Summary of the Numerical Example

1. **Initial WLDM Execution:**
   - Constructed the feature matrix $\mathbf{S}$ and target vector $\mathbf{y}$ with nonlinear basis functions.
   - Computed $\mathbf{S}^\top\mathbf{S}$ and $\mathbf{S}^\top\mathbf{y}$.
   - Estimated initial coefficients $\mathbf{A}^{(1)} = [1.7180, -0.1084]^\top$.
   - Calculated residuals $\mathbf{z}^{(1)} = \{-0.3904, 0.0024, -0.8216, 0.1376, 1.2350\}$.
   - Updated weights $p_t = \{0.8679, 0.999994, 0.5968, 0.9815, 0.3960\}$ to downweight larger residuals.
2. **Iterative GLDM Execution:**
   - Repeated the WLDM process with updated weights.
   - After one iteration, achieved convergence with final coefficients $\mathbf{A}^* = [1.75, -0.125]^\top$.

This comprehensive example illustrates how WLDM and GLDM collaboratively enhance model robustness and forecasting accuracy by effectively handling outliers and iteratively refining model parameters.

10 Differences Between GLDM and WLDM Algorithms

The **Weighted Least Deviation Method (WLDM)** and the **Generalized Least Deviation Method (GLDM)** are both designed to produce robust parameter estimates in the presence of noise and outliers. Despite their shared goal of minimizing deviations between observed values and model predictions, they differ in their objective functions, weight update mechanisms, nonlinearity modeling, and iterative refinement processes.

10.1 Key Differences

1. **Objective Function:** WLDM focuses on minimizing weighted absolute deviations as shown in Equation 10.1:

$$\min_{\mathbf{A},\mathbf{z}} \sum_{t=2}^{T} p_t |z_t|, \tag{10.1}$$

where $\mathbf{A}$ is the coefficient vector, $\mathbf{z}$ is the residual vector, and p_t are observation-specific weights. This linear-like approach primarily addresses outliers by assigning them lower weights but does not inherently model nonlinear relationships.

In contrast, GLDM employs an arctan-based loss function as shown in Equation 10.2:

$$\min_{a_1,a_2} \sum_{t=2}^{T} \arctan\left(\left|a_1 \cdot y_{t-1} + a_2 \cdot y_{t-1}^2 - y_t\right|\right), \tag{10.2}$$

where a_1 and a_2 are coefficients and y_{t-1} is the basis function. The arctan function grows slowly for large residuals, naturally reducing the influence of extreme outliers and allowing modeling of more complex, nonlinear structures.

2. **Weight Updates:** In WLDM, the weights p_t are generally fixed or determined from a preliminary analysis. There is no inherent iterative adjustment of these weights once the model is fitted.

GLDM, however, updates the weights at each iteration to further downweight large deviations. This update rule, shown in Equation 10.3, is:

$$p_t = \frac{1}{1 + (z_t^{(k)})^2}, \quad \forall t, \tag{10.3}$$

where $z_t^{(k)}$ is the residual at iteration k. This iterative reweighting process ensures that points with large residuals in one iteration have reduced influence in the next, leading to a more stable and refined solution.

3. **Nonlinearity and Basis Functions:** WLDM typically assumes a linear model structure, as indicated by Equation 10.4:

$$z_t = a_1 \cdot y_{t-1} + a_2 \cdot y_{t-1}^2 - y_t. \tag{10.4}$$

   This linear assumption can limit its ability to capture complex dynamics.

   GLDM, by contrast, accommodates nonlinear relationships through quasilinear basis functions. For a first-order model ($m = 1$), the basis functions include:

$$g_1(y_{t-1}) = y_{t-1}, \quad \text{(Linear term)} \tag{10.5}$$
$$g_2(y_{t-1}) = y_{t-1}^2, \quad \text{(Quadratic term)} \tag{10.6}$$

   Such flexibility allows GLDM to capture complex patterns that WLDM may not model effectively.

4. **Iterative Refinement:** WLDM is generally a one-shot procedure. Once the coefficients $\mathbf{A}$ are computed to minimize Equation 10.1, there is no built-in mechanism for iterative refinement.

   In GLDM, the solution is refined through multiple iterations. After each iteration, the weights and possibly the basis functions are updated, and WLDM is solved again with these updated parameters. As shown in Equation 10.7:

$$(\mathbf{A}^{(k+1)}, \mathbf{z}^{(k+1)}) = \text{WLDM}(S, \{p_t^{(k)}\}, Y), \tag{10.7}$$

   where $\mathbf{A}^{(k+1)}$ and $\mathbf{z}^{(k+1)}$ denote updated coefficients and residuals at iteration $k+1$. This iterative process continues until convergence, resulting in a solution that is both robust and better tuned to the underlying data structure.

5. **Sensitivity to Outliers:** WLDM provides moderate sensitivity to outliers through fixed or predefined weights. GLDM minimizes sensitivity to outliers by iteratively adjusting weights based on residuals, ensuring that extreme deviations have minimal impact on the final model.

10.2 Numerical Example

Consider the time series:

$$y = \{1, 2, 3, 4, 5, 6\},$$

where $y_2 = 2$, $y_3 = 3$, $y_4 = 4$, $y_5 = 5$, and $y_6 = 6$ are the target values to be predicted using the preceding observation y_{t-1}.
WLDM Approach: Using WLDM (Equation 10.1) with uniform weights $p_t = 1$, the model finds coefficients that fit the data by minimizing the sum of weighted absolute deviations.
GLDM Approach: Applying GLDM (Equation 10.2) initially sets $p_t = 1$. After each iteration, weights corresponding to larger residuals are significantly reduced using Equation 10.3. This iterative process continues until convergence, reducing the influence of outliers and refining the coefficient estimates.

11 Summary of Differences

Table 11.1 summarizes the key differences between WLDM and GLDM, referencing the relevant equations discussed above.

Feature	**WLDM**	**GLDM**
Objective Function	Weighted absolute deviations (Eq. 10.1)	Arctan-based loss (Eq. 10.2)
Weight Updates	Fixed or predefined	Iterative (Eq. 10.3)
Nonlinearity	Primarily linear (Eq. 10.4)	Nonlinear basis (Eqs. 10.5–10.6)
Iterative Refinement	Not inherent	Yes (Eq. 10.7)
Sensitivity to Outliers	Moderate	Minimal (due to iterative reweighting)

Table 11.1: Comparison of WLDM and GLDM Features

WLDM provides a robust yet generally linear modeling framework, while GLDM extends this robustness with iterative weight updates and nonlinear basis functions. This additional complexity enables GLDM to better handle outliers and nonlinearities, resulting in more accurate and stable model estimates.

12 Visual Representations of WLDM, GLDM, and Predictor Algorithm

In this chapter, we present six figures that visually illustrate the workflows and results of the Weighted Least Deviation Method (WLDM), Generalized Least Deviation Method (GLDM), and the Predictor Algorithm. Each figure is designed to enhance understanding by providing clear, professional visualizations of each process and their respective outcomes.

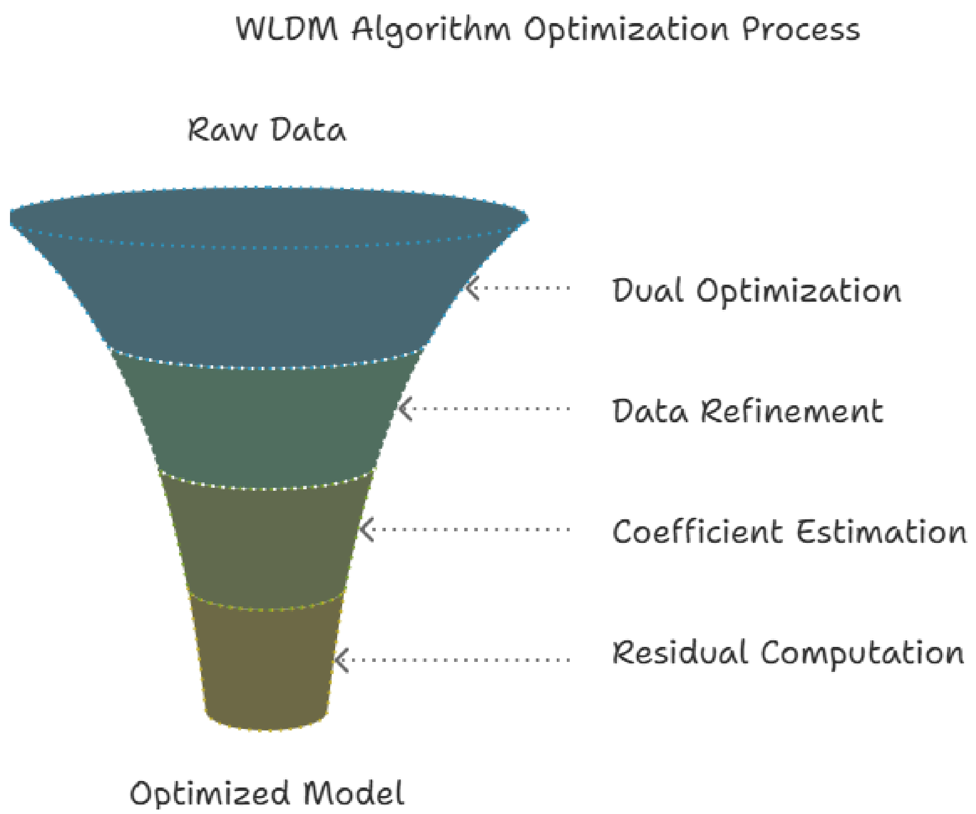

Figure 12.1: Workflow of the Weighted Least Deviation Method (WLDM) Algorithm

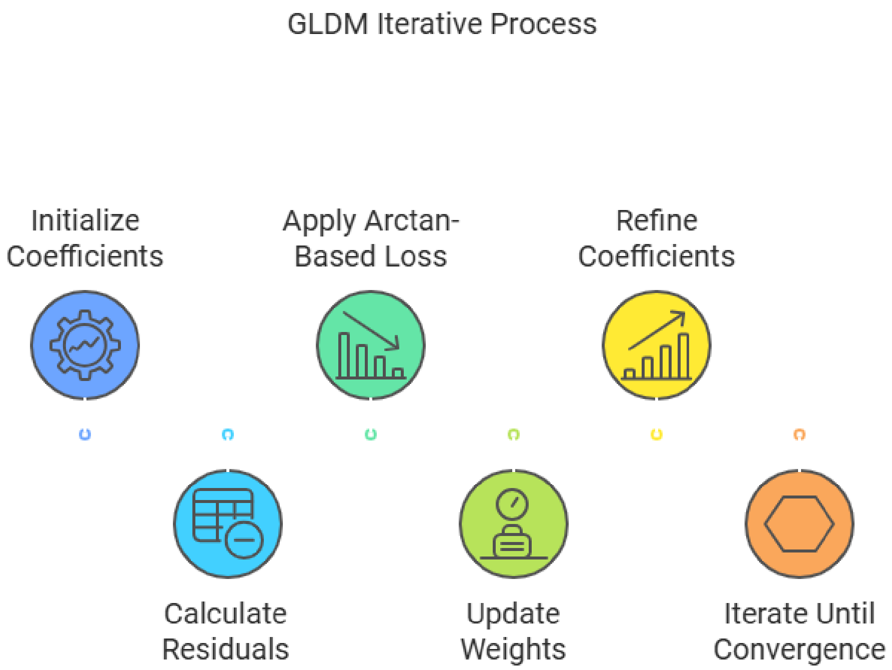

Figure 12.2: Initial Coefficient Estimation in WLDM

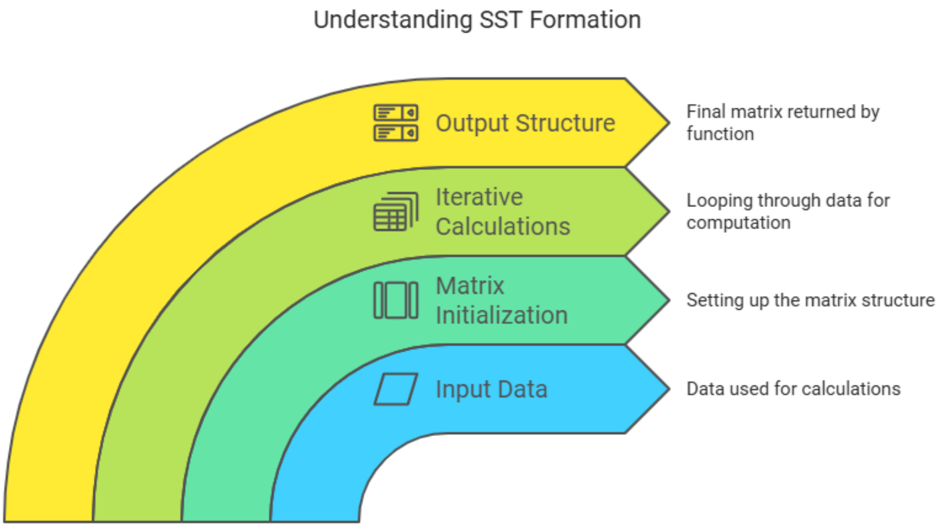

Figure 12.3: Residual Computation Process in WLDM

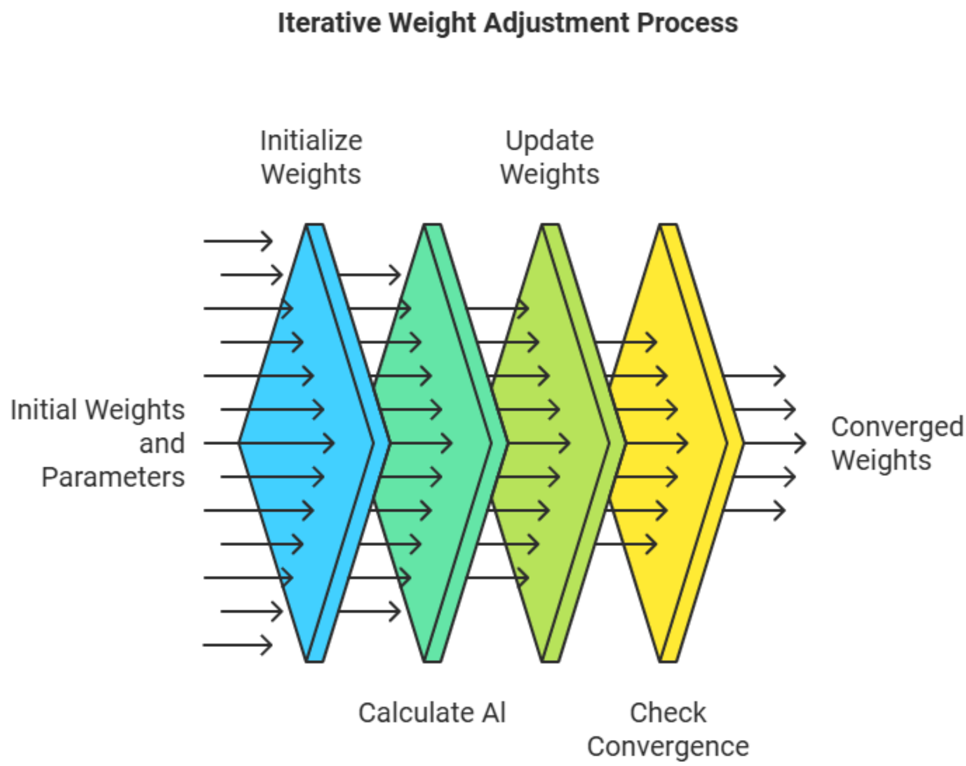

Figure 12.4: Weight Update Mechanism in WLDM

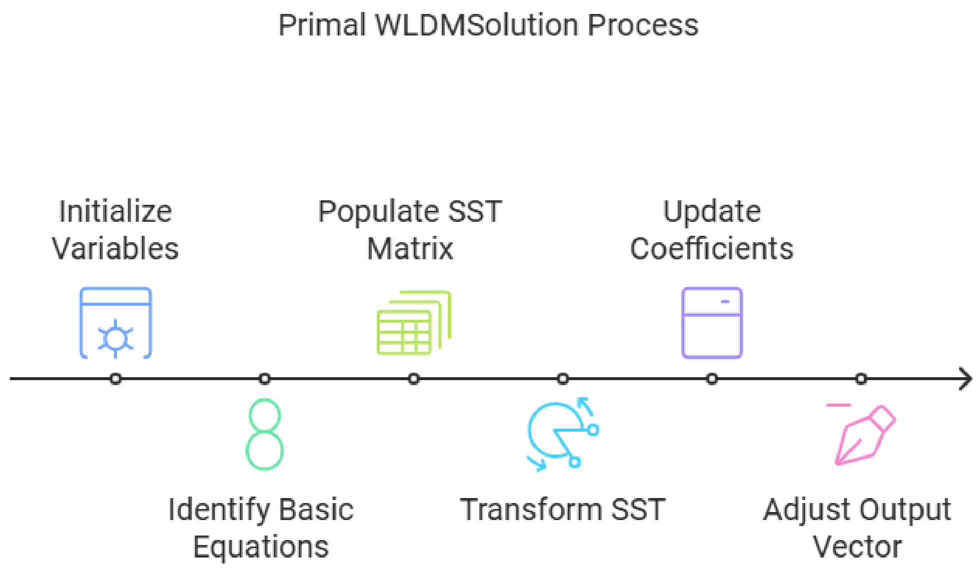

Figure 12.5: Iterative Refinement Process of the Generalized Least Deviation Method (GLDM)

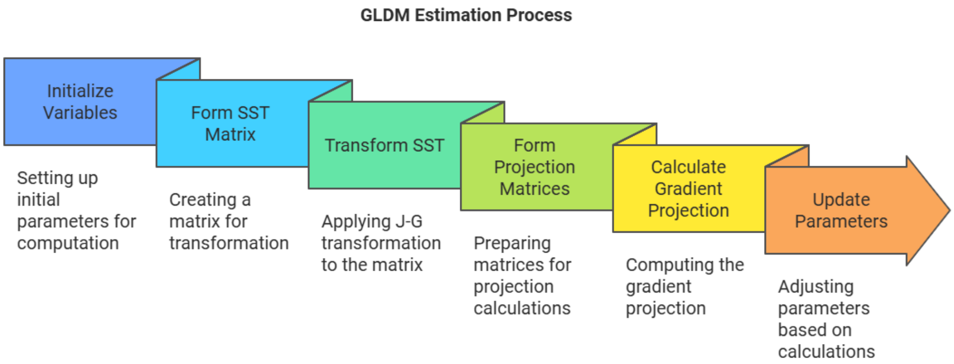

Figure 12.6: Structure of the Predictor Algorithm

12.1 Figure Descriptions

Figure 12.1 illustrates the comprehensive workflow of the Weighted Least Deviation Method (WLDM) algorithm. This flowchart outlines the sequential steps, including data preprocessing, construction of input matrices, computation of residuals, and the application of weight adjustments to mitigate the influence of outliers. The diagram emphasizes how WLDM systematically minimizes weighted deviations to achieve robust coefficient estimation.

Figure 12.2 depicts the initial coefficient estimation phase within WLDM. It showcases how the algorithm leverages the constructed input matrices to compute preliminary model coefficients. This foundational step establishes the initial relationship between input features and the target variable before any iterative refinement processes are undertaken.

Figure 12.3 illustrates the residual computation process in WLDM. After obtaining the initial coefficients, the algorithm calculates residuals, which represent the differences between observed values and model predictions. This step is crucial for identifying outliers that may disproportionately influence the model.

Figure 12.4 showcases the weight update mechanism employed by WLDM. Based on the computed residuals, the algorithm adjusts the weights p_t assigned to each observation. Observations with larger residuals receive smaller weights, effectively reducing their impact on subsequent iterations and enhancing the model's robustness against outliers.

Figure 12.5 visualizes the iterative refinement process of the Generalized Least Deviation Method (GLDM). Building upon WLDM, GLDM repeatedly updates weights and recalculates coefficients through successive iterations. This iterative process allows GLDM to better handle complex, nonlinear relationships within the data and further mitigate the influence of outliers.

Figure 12.6 presents the structure of the Predictor Algorithm. This figure outlines how the algorithm utilizes the optimized coefficients from WLDM and GLDM to generate future forecasts. It highlights the integration of error thresholding mechanisms, which assess and ensure the reliability of the predictions by stopping the forecasting process when errors exceed predefined limits.

12.2 Summary

The figures collectively provide a comprehensive visual summary of the WLDM, GLDM, and Predictor Algorithm methodologies:

- **Figure 12.1**: Details the overall workflow of WLDM.
- **Figure 12.2**: Shows the initial estimation of model coefficients in WLDM.
- **Figure 12.3**: Illustrates the computation of residuals in WLDM.
- **Figure 12.4**: Demonstrates the weight updating process to handle outliers in WLDM.
- **Figure 12.5**: Depicts the iterative refinement steps in GLDM.
- **Figure 12.6**: Outlines the Predictor Algorithm's forecasting structure.

These visual aids enhance the understanding of robust optimization techniques and their practical implementations in time series forecasting, providing clear and professional illustrations of each step and their respective impacts on model performance.