Methods of Interpolation

A Comparative Analysis to Identify Optimal Techniques for Mining Trajectory Data

Caleb Koch

Geospatial Research Laboratory

July 2016

Problem Statement

Spatiotemporal trajectory data often consist of gaps and missing points due to errors in the collection procedure.

Researchers employ interpolation methods to fill in these gaps.

Lots of different interpolation methods exist for different applications.

The problem is to identify and assess the interpolation methods that could potentially work with trajectories mined using mSEQUITUR.

Problem Statement

Spatiotemporal trajectory data often consist of gaps and missing points due to errors in the collection procedure.

Researchers employ interpolation methods to fill in these gaps.

Lots of different interpolation methods exist for different applications.

The problem is to identify and assess the interpolation methods that could potentially work with trajectories mined using mSEQUITUR.

Problem Statement

Spatiotemporal trajectory data often consist of gaps and missing points due to errors in the collection procedure.

Researchers employ interpolation methods to fill in these gaps.

Lots of different interpolation methods exist for different applications.

The problem is to identify and assess the interpolation methods that could potentially work with trajectories mined using mSEQUITUR.

Problem Statement

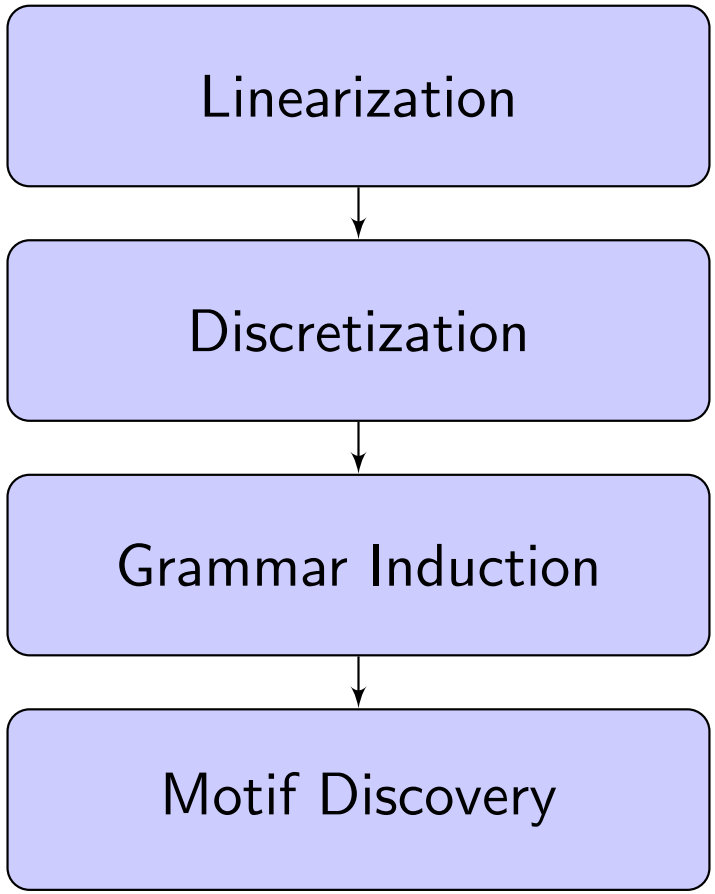
Spatiotemporal trajectory data often consist of gaps and missing points due to errors in the collection procedure.

Researchers employ interpolation methods to fill in these gaps.

Lots of different interpolation methods exist for different applications.

The problem is to identify and assess the interpolation methods that could potentially work with trajectories mined using mSEQUITUR.

Processing with mSEQUITUR

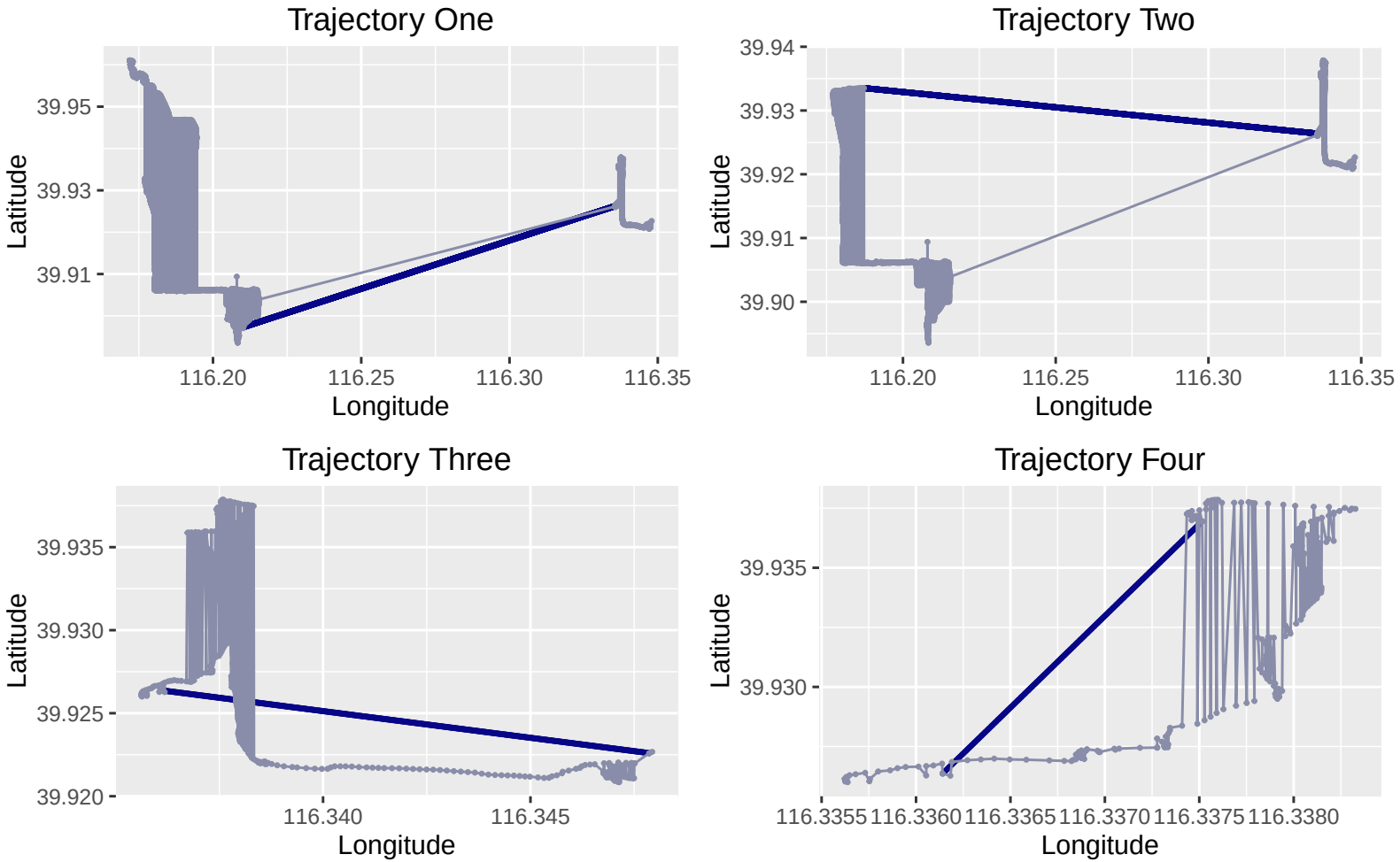


Considerations:

- Interpolation can take place at multiple steps
- Benefits and drawbacks of each
- Using motifs to obtain greater precision

Current Approach and Associated Issues

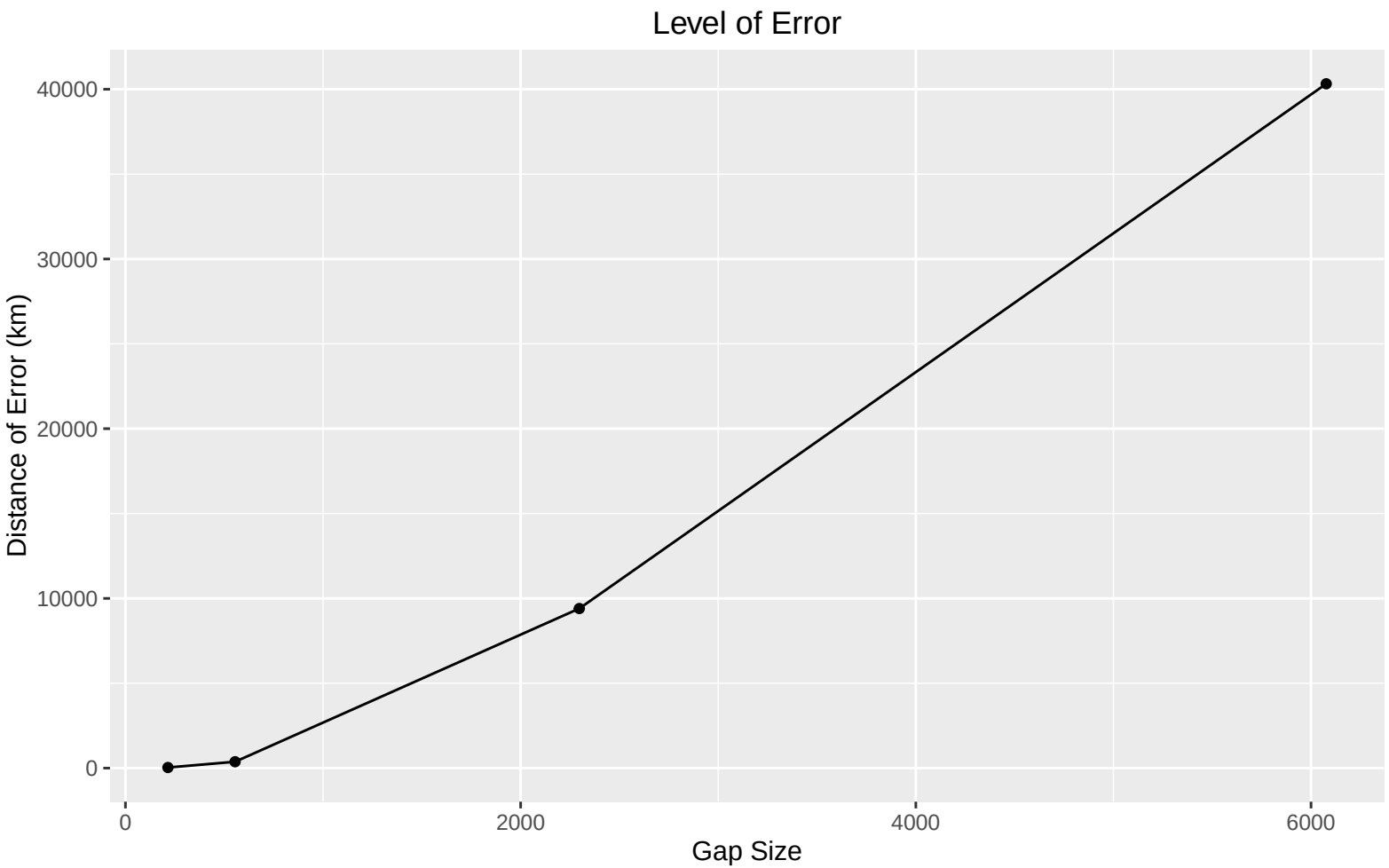
Current approach uses linear interpolation.



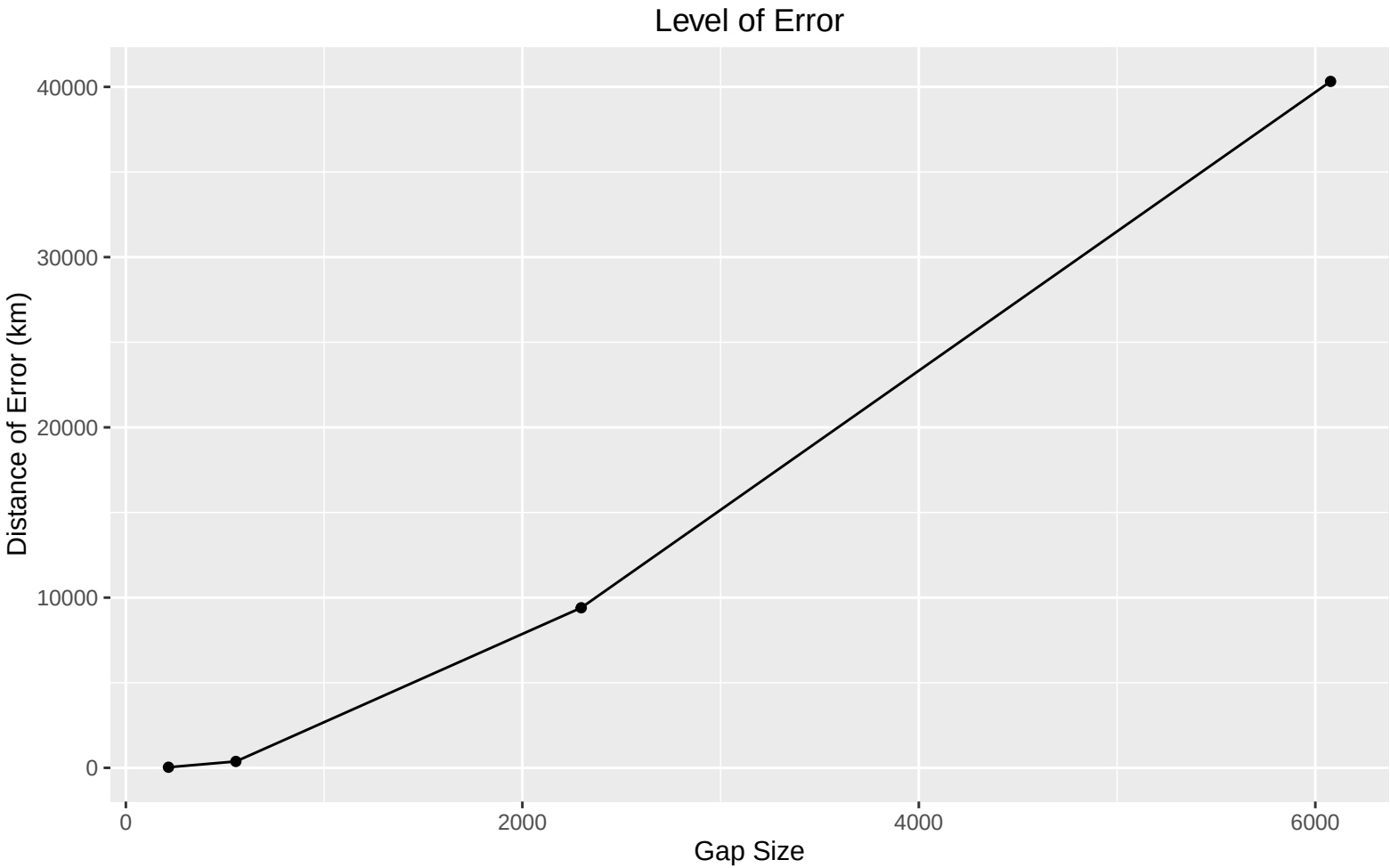
Linear interpolation appears highly innaccurate.

Summaries of Error

Number of points	Distance (km) of error	Distance (km) traveled	Duration (hours)	Error (km) per min
6077	40322.53968	59.62520373	8.83	76.10898391
2297	9403.933131	35.53018522	3.31	47.35112352
555	375.9281143	5.062658871	0.51	12.28523249
215	35.42083656	1.742807734	0.22	2.683396709



Summaries of Error



Assuming a linear relationship, there is an increase of approximately 7 kilometers for each interpolated point.

Characteristics of an Interpolation Method

An interpolation method is defined by a series of unique characteristics including

- Local/global
- Exact/inexact
- Deterministic/stochastic
- Complexity

Throughout our analysis, we consider which characteristics are more suitable than others

Potential Methods

- Barnes interpolation
- Spline interpolation
- Polynomial Curve Fitting
- Bilinear
- Kriging
- Principal curve detection (Biau & Fischer, 2012)
- Frequent trajectory mining (Savage et al., 2010)
- Linear interpolation
- Neural network/machine learning
- Probabilistic modeling
- Nearest Neighbor
- Natural Neighbor
- Inverse distance weighting
- Latent statistical model (Kishonishita et al., 2016)
- Dynamic time warping-based similarity (Sankararaman et al., 2013)

Potential Methods

- Barnes interpolation
- Spline interpolation
- Polynomial Curve Fitting
- Bilinear
- Kriging
- Principal curve detection (Biau & Fischer, 2012)
- Frequent trajectory mining (Savage et al., 2010)
- Linear interpolation
- Neural network/machine learning
- Probabilistic modeling
- Nearest Neighbor
- Natural Neighbor
- Inverse distance weighting
- Latent statistical model (Kishonishita et al., 2016)
- Dynamic time warping-based similarity (Sankararaman et al., 2013)

Testing Interpolation Methods

We need a standard framework for assessing accuracy.

General outline:

1. Select a sample group of trajectories representing a range of features (i.e. length)

Testing Interpolation Methods

We need a standard framework for assessing accuracy.

General outline:

1. Select a sample group of trajectories representing a range of features (i.e. length)
2. Generate a range of artificial gaps

Testing Interpolation Methods

We need a standard framework for assessing accuracy.

General outline:

1. Select a sample group of trajectories representing a range of features (i.e. length)
2. Generate a range of artificial gaps
3. Use interpolation method to fill in the gaps

Testing Interpolation Methods

We need a standard framework for assessing accuracy.

General outline:

1. Select a sample group of trajectories representing a range of features (i.e. length)
2. Generate a range of artificial gaps
3. Use interpolation method to fill in the gaps
4. Run error analysis

Testing Interpolation Methods

We need a standard framework for assessing accuracy.

General outline:

1. Select a sample group of trajectories representing a range of features (i.e. length)
2. Generate a range of artificial gaps
3. Use interpolation method to fill in the gaps
4. Run error analysis
5. Compare errors by calculating representative inaccuracies

Selecting trajectories

1. Select a sample group of trajectories representing a range of features (i.e. length)

Geolife dataset offers a record of GPS trajectories for more than 150 users over four years.

Selecting trajectories

1. Select a sample group of trajectories representing a range of features (i.e. length)

Geolife dataset offers a record of GPS trajectories for more than 150 users over four years. Points are normally recorded every 5 seconds.

Selecting trajectories

1. Select a sample group of trajectories representing a range of features (i.e. length)

Geolife dataset offers a record of GPS trajectories for more than 150 users over four years. Points are normally recorded every 5 seconds. Trajectories range in size from less than 10 points (less than 30 seconds) to more than 45,000 (more than 24 hours).

Selecting trajectories

1. Select a sample group of trajectories representing a range of features (i.e. length)

Geolife dataset offers a record of GPS trajectories for more than 150 users over four years. Points are normally recorded every 5 seconds. Trajectories range in size from less than 10 points (less than 30 seconds) to more than 45,000 (more than 24 hours). Our selection includes

- 20 trajectories
- Variety acheived through file size
- Largest trajectory is $\approx$ 3480 KB, smallest is $\approx$ 15 KB
- Every other is within roughly $\pm$100 KB of the next largest/smallest file

Creation of Gaps

2. Generate a range of artificial gaps

Trajectory Representation

Denote the 20 trajectories as a set $T = \{t_1, t_2, ..., t_{20}\}$ where t is a trajectory and

$$|t_{n-1}| < |t_n| < |t_{n+1}|, \forall\ t \in T.$$

Let g_n give the gap length for trajectory t_n. The value of g_n depends on the size of trajectory t_n.

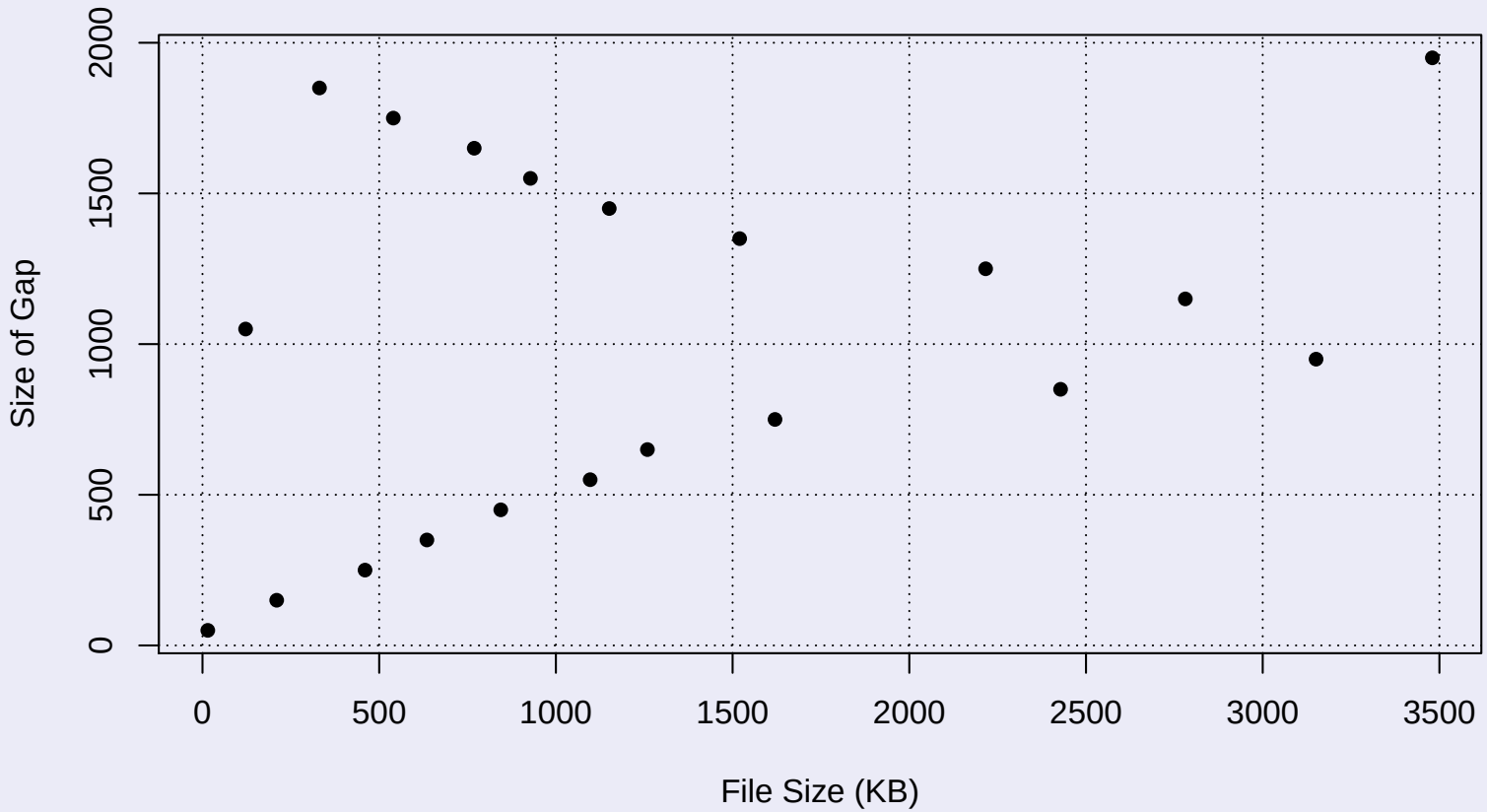
Selection of g_n

The following algorithm summarizes the selection of g_n.

1: $n \leftarrow 1$
2: **while** $n \leq 20$ **do**
3: **if** $n = 1$ **then**
4: $g_n \leftarrow 50$ ▷ smallest gap size
5: **else if** $n = 20$ **then**
6: $g_n \leftarrow 1950$ ▷ largest gap size
7: **else if** $n \pmod 2 \equiv 0$ **then**
8: $g_n = 50n + 50$
9: **else**
10: $g_n = 2000 - 50n$
11: **end if**
12: $n \leftarrow n + 1$
13: **end while**

Visualizing the gap length selection

Gap Sizes



Applying Interpolation

3. Use interpolation method to fill in the gaps

Requires an implementation of the method.

Some implementations suitable for this particular type of analysis exist and are accessible; others are not.

Create a new trajectory t' for each original trajectory.

Error Analysis

4. Run error analysis

Total error for a particular dataset is denoted e_n. We can determine e_n by the following.

- Consider a point $p_i \in t_n$
- For each p_i, there is a corresponding $p_i' \in t_n'$
- Let d_i denote the distance between p_i and p_i'
- Then $e_n = \sum d_i$

Error Analysis

4. Run error analysis

Total error for a particular dataset is denoted e_n. We can determine e_n by the following.

- Consider a point $p_i \in t_n$
- For each p_i, there is a corresponding $p_i' \in t_n'$
- Let d_i denote the distance between p_i and p_i'
- Then $e_n = \sum d_i$

Note that e_n maintains the units of the original dataset, so it cannot be used to compare error from time series interpolation and trajectory interpolation.

Comparative Analysis

5. Compare errors by calculating representative inaccuracies

In order to compare e_n values between time series and trajectories, we introduce e_s, the standard error where

$$e_s = \frac{e_n}{d_s}.$$

Here, d_s is the standard distance and is calculated from the original trajectory or time series.

$$d_s = \sum \text{dist}(p_i, p_{i+1})$$

Comparative Analysis

5. Compare errors by calculating representative inaccuracies

In order to compare e_n values between time series and trajectories, we introduce e_s, the standard error where

$$e_s = \frac{e_n}{d_s}.$$

Here, d_s is the standard distance and is calculated from the original trajectory or time series.

$$d_s = \sum \text{dist}(p_i, p_{i+1})$$

The standard error is unitless and can be compared across interpolation methods used at different stages of pre-processing.

Nearest Neighbor Overview

- Perhaps the most fundamental method of interpolation

Nearest Neighbor Overview

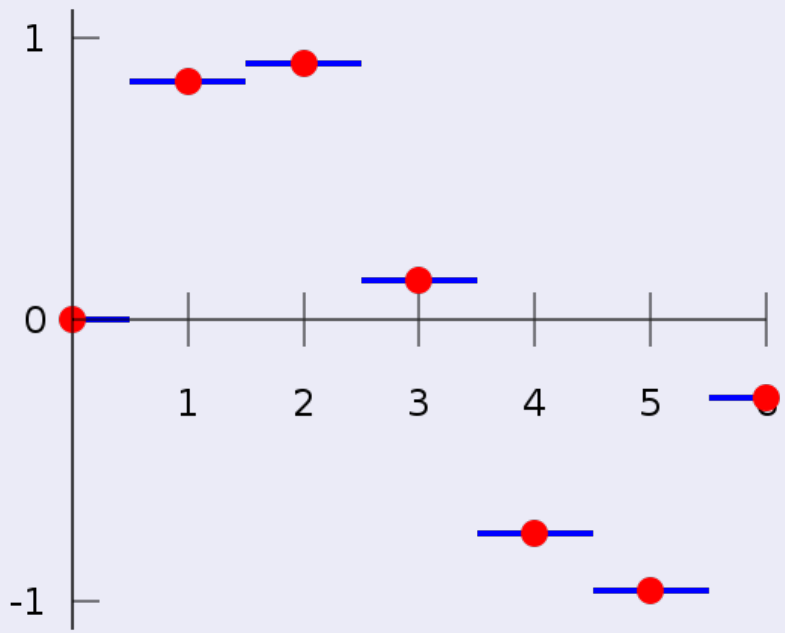
- Perhaps the most fundamental method of interpolation
- Involves copying the information from known data points for creating interpolated points

Nearest Neighbor Overview

- Perhaps the most fundamental method of interpolation
- Involves copying the information from known data points for creating interpolated points
- This method can be applied to both spatial trajectory data and time series data

Nearest Neighbor Overview

- Perhaps the most fundamental method of interpolation
- Involves copying the information from known data points for creating interpolated points
- This method can be applied to both spatial trajectory data and time



en.wikipedia.org/wiki/Nearest-neighbor_interpolation

Formal Definition

Suppose the gap of interest for interpolating s points exists between (x_i, y_i) and (x_{i+1}, y_{i+1}).

The desired set of interpolated points is denoted as $I = \{(a_1, b_1), (a_2, b_2), ..., (a_s, b_s)\}$.

Then, nearest neighbor interpolation involves computing I according to

$$b_n = y_i, \ \forall \ n \in [1, \lfloor s/2 \rceil] \subset \mathbb{Z}$$
$$b_n = y_{i+1}, \ \forall \ n \in (\lfloor s/2 \rceil, s] \subset \mathbb{Z}$$
$$a_n = a_{n-1} + \frac{x_{i+1} - x_i}{s+1}, \forall \ n \in [2, s] \subset \mathbb{Z}$$

such that $a_1 = x_i + {}^{(x_{i+1}-x_i)}/_{(s+1)}$.

With the time series data, y would denote the Hilbert value while x would denote the corresponding index.

Results

Trajectory

- The sum of errors (i.e. $\sum e_n$): 98338.29804 km
- Sum of gap lengths: 314.6696176 km
- The sum of errors squared: 1617790.129 km
- $\sum e_s/20 = 176.2701154$ (Average e_s)

Time Series

- The sum of errors (i.e. $\sum e_n$): 577743 Hilberts
- Sum of gap lengths: 21519.8334 Hilberts
- The sum of errors squared: 36426269 Hilberts
- $\sum e_s/20 = 18.04700582$ (Average e_s)

Results

Trajectory

- The sum of errors (i.e. $\sum e_n$): 98338.29804 km
- Sum of gap lengths: 314.6696176 km
- The sum of errors squared: 1617790.129 km
- $\sum e_s/20 = 176.2701154$ (Average e_s)

Time Series

- The sum of errors (i.e. $\sum e_n$): 577743 Hilberts
- Sum of gap lengths: 21519.8334 Hilberts
- The sum of errors squared: 36426269 Hilberts
- $\sum e_s/20 = 18.04700582$ (Average e_s)

Results

Trajectory

- The sum of errors (i.e. $\sum e_n$): 98338.29804 km
- Sum of gap lengths: 314.6696176 km
- The sum of errors squared: 1617790.129 km
- $\sum e_s/20 = 176.2701154$ (Average e_s)

Time Series

- The sum of errors (i.e. $\sum e_n$): 577743 Hilberts
- Sum of gap lengths: 21519.8334 Hilberts
- The sum of errors squared: 36426269 Hilberts
- $\sum e_s/20 = 18.04700582$ (Average e_s)

Initial Observations

- Interpolating time series reduced error considerably
- Error tends to increase with gap length
- Smaller gap length showed little difference in error (preprocessing stage less important)

Linear Interpolation Overview

- Most common method of interpolation

Linear Interpolation Overview

- Most common method of interpolation
- Constructs a line connecting known points

Linear Interpolation Overview

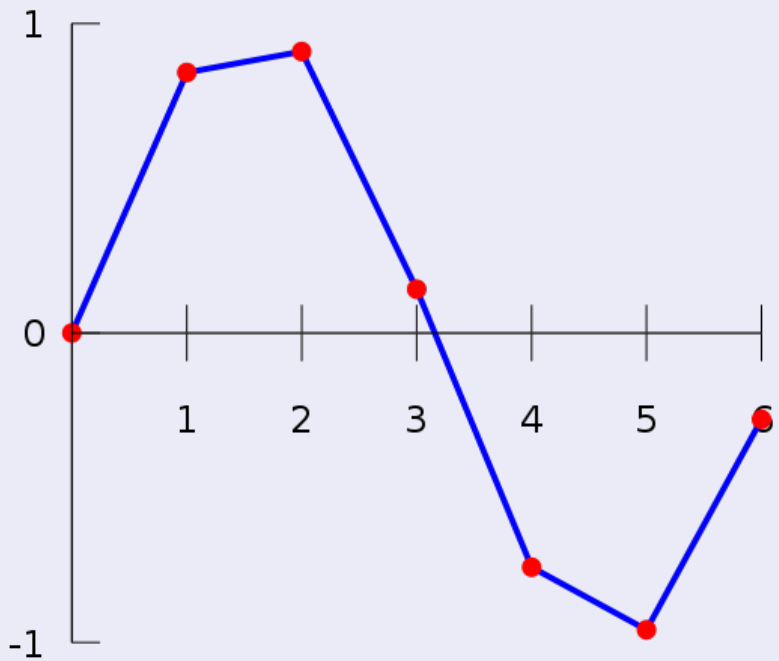
- Most common method of interpolation
- Constructs a line connecting known points
- This method can be applied to both spatial trajectory data and time series data

Linear Interpolation Overview

- Most common method of interpolation
- Constructs a line connecting known points
- This method can be applied to both spatial trajectory data and time series data
- Easy to implement and currently used as default

Linear Interpolation Overview

- Most common method of interpolation
- Constructs a line connecting known points
- This method can be applied to both spatial trajectory data and time series data



en.wikipedia.org/wiki/Linear_interpolation

Formal Definition

Given two known points (x_i, y_i) and (x_{i+1}, y_{i+1}) between which exists the gap to be interpolated, the interpolant is

$$b_n = \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)(a_n - x_i) + y_i, \ \forall \ n \in [1, s] \subset \mathbb{Z}$$

$$a_n = a_{n-1} + \frac{x_{i+1} - x_i}{s+1}, \forall \ n \in [2, s] \subset \mathbb{Z}$$

where $a_1 = x_i + {}^{(x_{i+1}-x_i)}/_{(s+1)}$.

Remember that the desired set of interpolated points is $I = \{(a_1, b_1), (a_2, b_2), ..., (a_s, b_s)\}$. Note that s is determined from g_n.

Results

Trajectory

- The sum of errors (i.e. $\sum e_n$): 42321.76999 km
- Sum of gap lengths: 314.6696176 km
- The sum of errors squared: 190500.3207 km
- $\sum e_s/20 = 110.2836019$ (Average e_s)

Time Series

- The sum of errors (i.e. $\sum e_n$): 605145.58394155 Hilberts
- Sum of gap lengths: 21519.8334 Hilberts
- The sum of errors squared: 31128429.9897611 Hilberts
- $\sum e_s/20 = 18.84167077$ (Average e_s)

Results

Trajectory

- The sum of errors (i.e. $\sum e_n$): 42321.76999 km
- Sum of gap lengths: 314.6696176 km
- The sum of errors squared: 190500.3207 km
- $\sum e_s/20 = 110.2836019$ (Average e_s)

Time Series

- The sum of errors (i.e. $\sum e_n$): 605145.58394155 Hilberts
- Sum of gap lengths: 21519.8334 Hilberts
- The sum of errors squared: 31128429.9897611 Hilberts
- $\sum e_s/20 = 18.84167077$ (Average e_s)

Results

Trajectory

- The sum of errors (i.e. $\sum e_n$): 42321.76999 km
- Sum of gap lengths: 314.6696176 km
- The sum of errors squared: 190500.3207 km
- $\sum e_s/20 = 110.2836019$ (Average e_s)

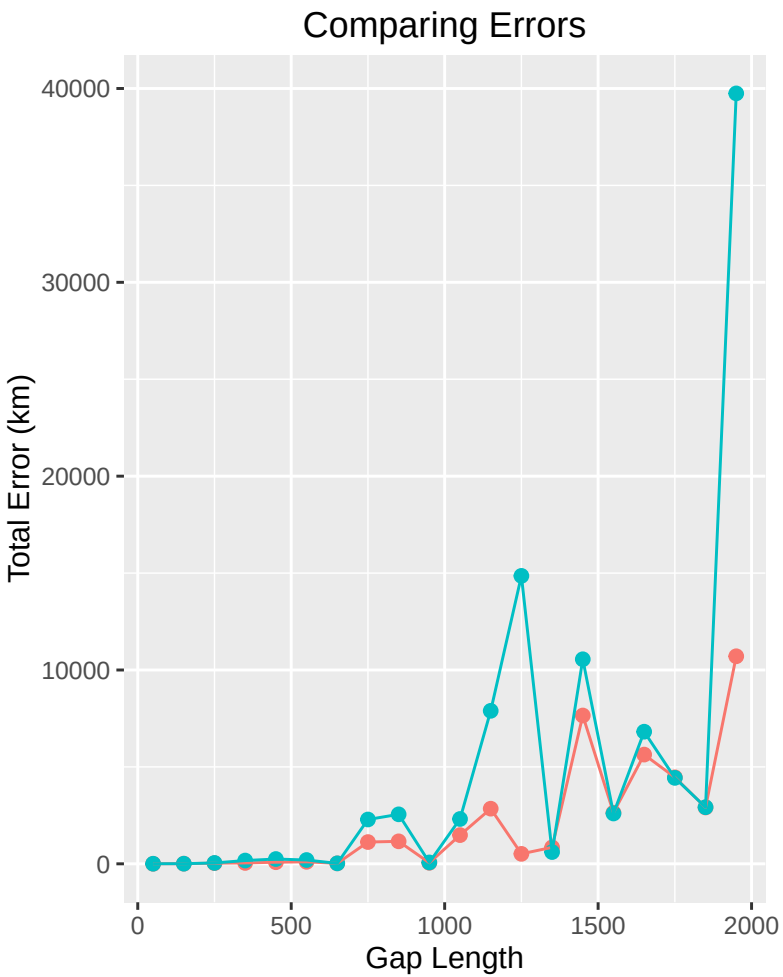
Time Series

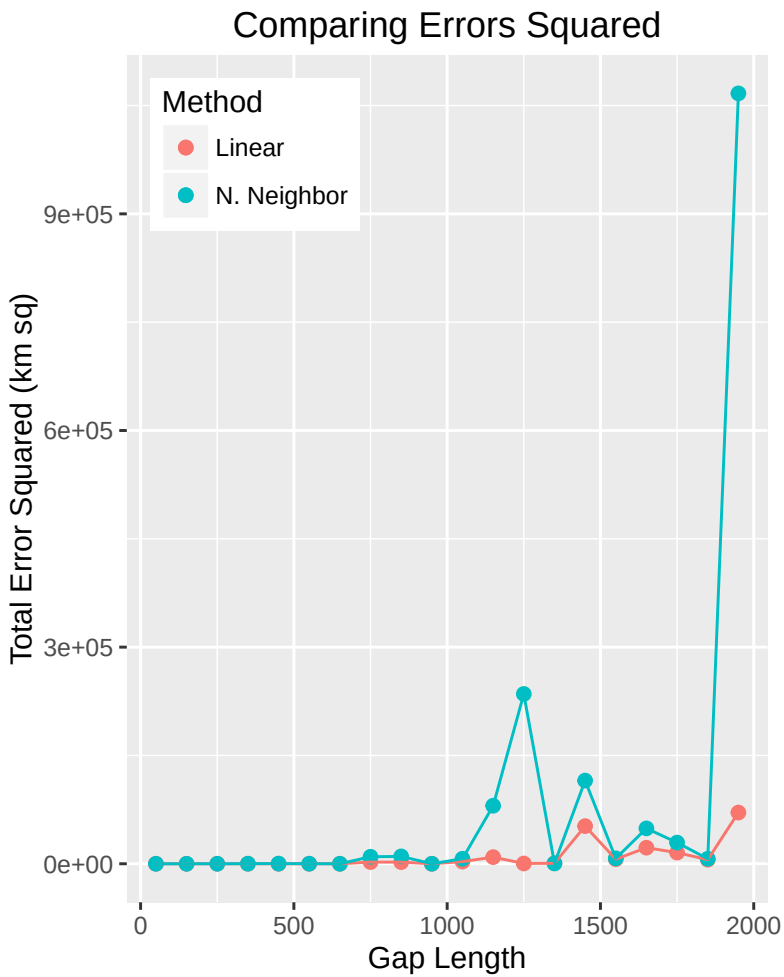
- The sum of errors (i.e. $\sum e_n$): 605145.58394155 Hilberts
- Sum of gap lengths: 21519.8334 Hilberts
- The sum of errors squared: 31128429.9897611 Hilberts
- $\sum e_s/20 = 18.84167077$ (Average e_s)

Initial Observations

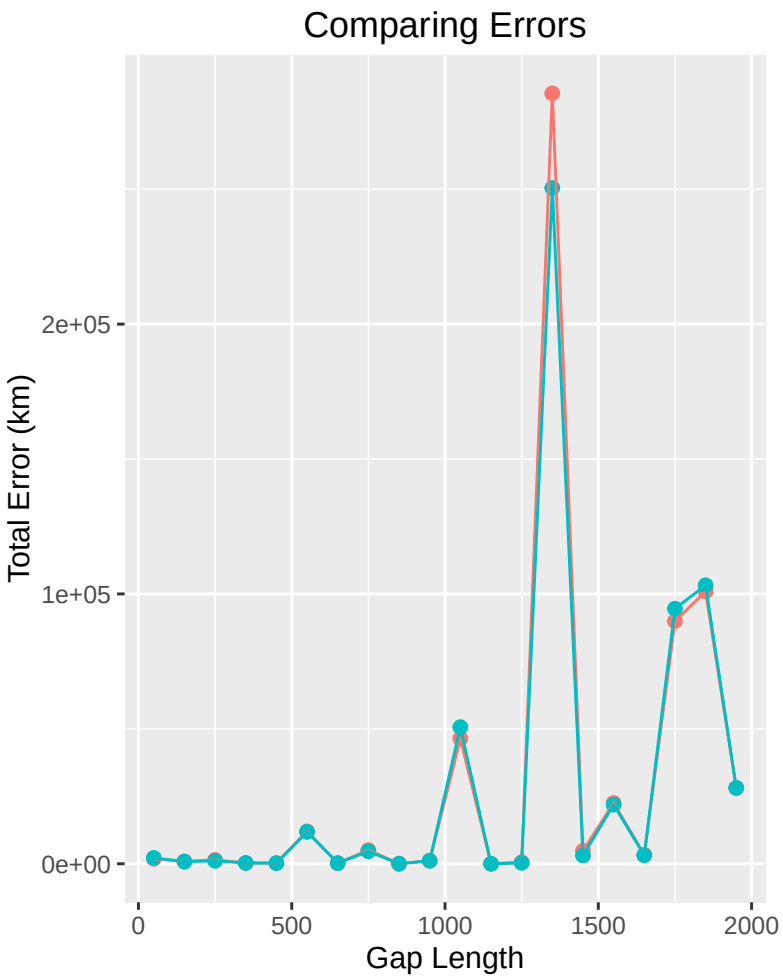
- Interpolating time series reduced error considerably
- Error tends to increase with gap length
- Smaller gap length showed little difference in error (preprocessing stage less important)

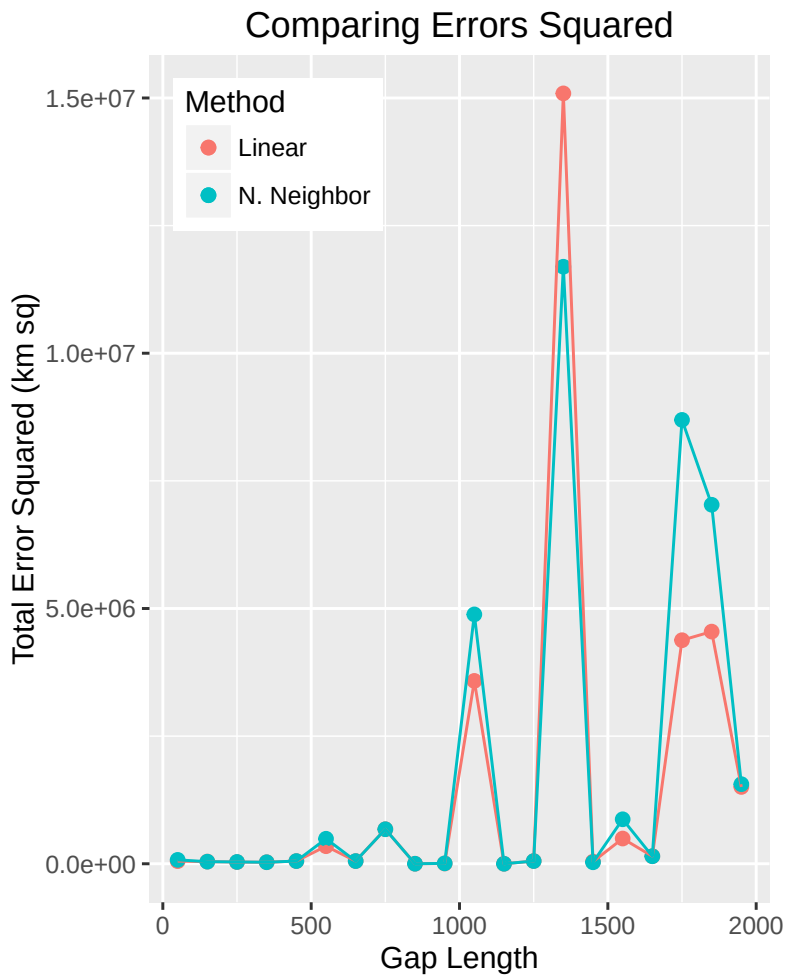
Combined Results: Trajectory Performance



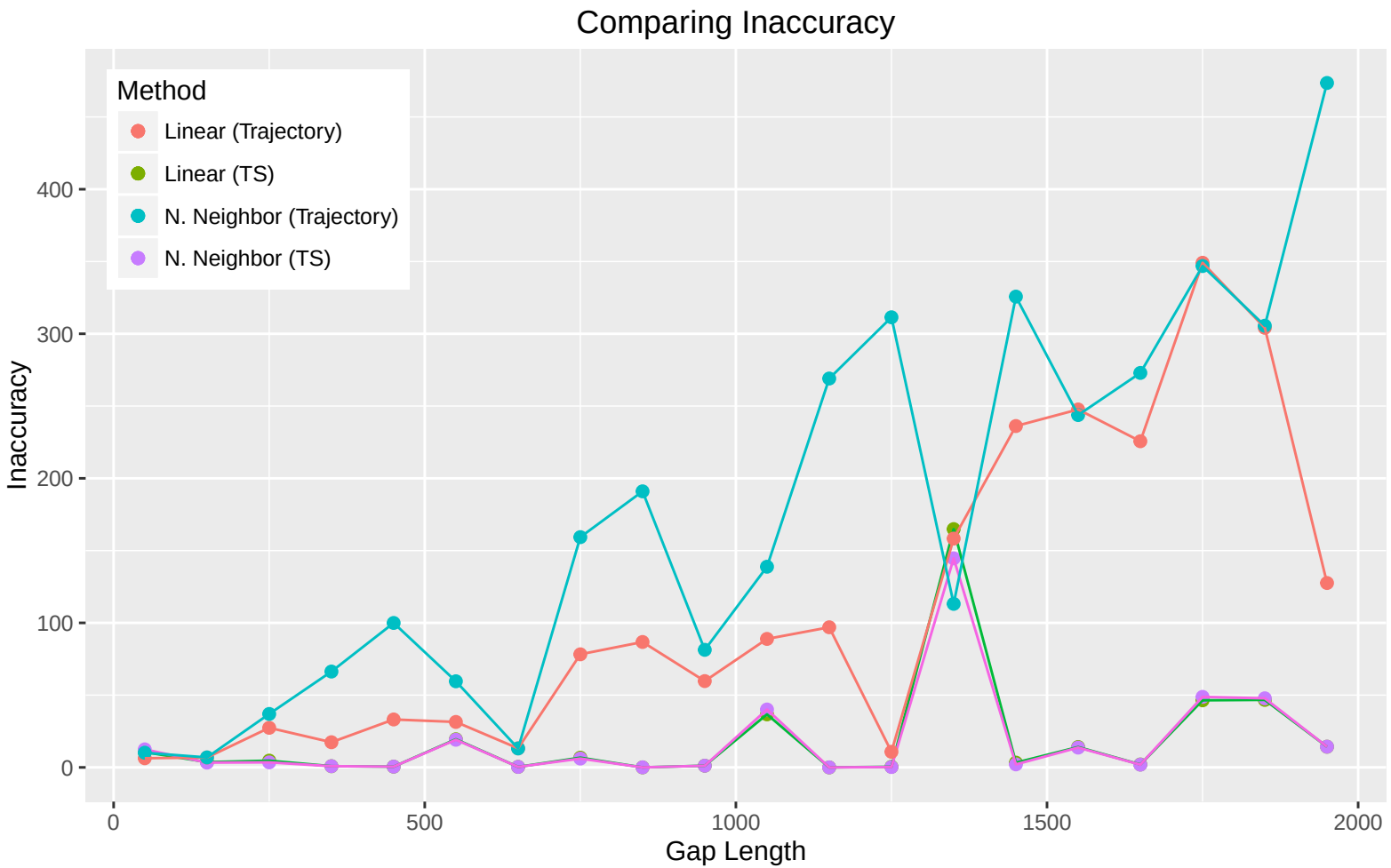


Combined Results: Time Series Performance

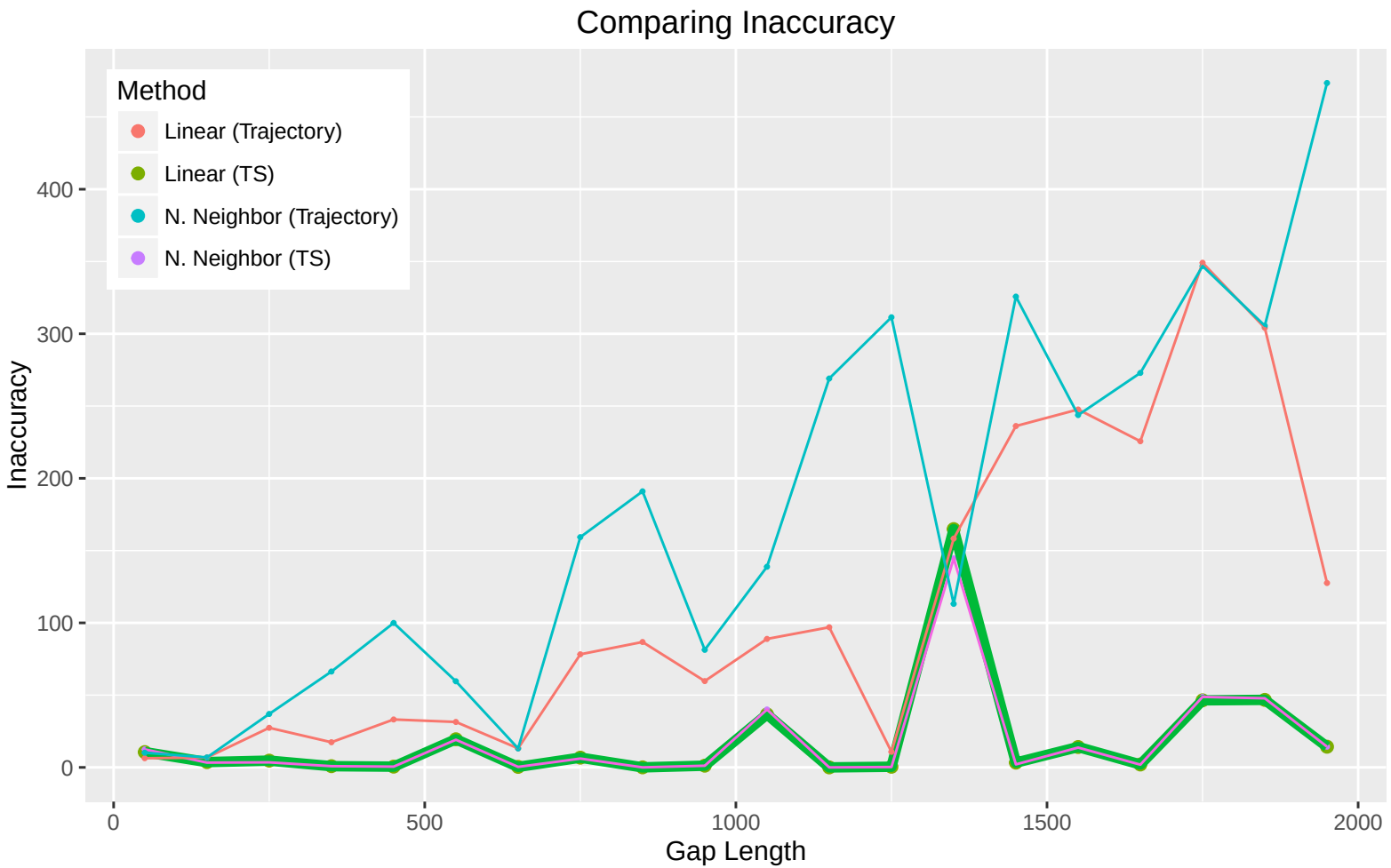




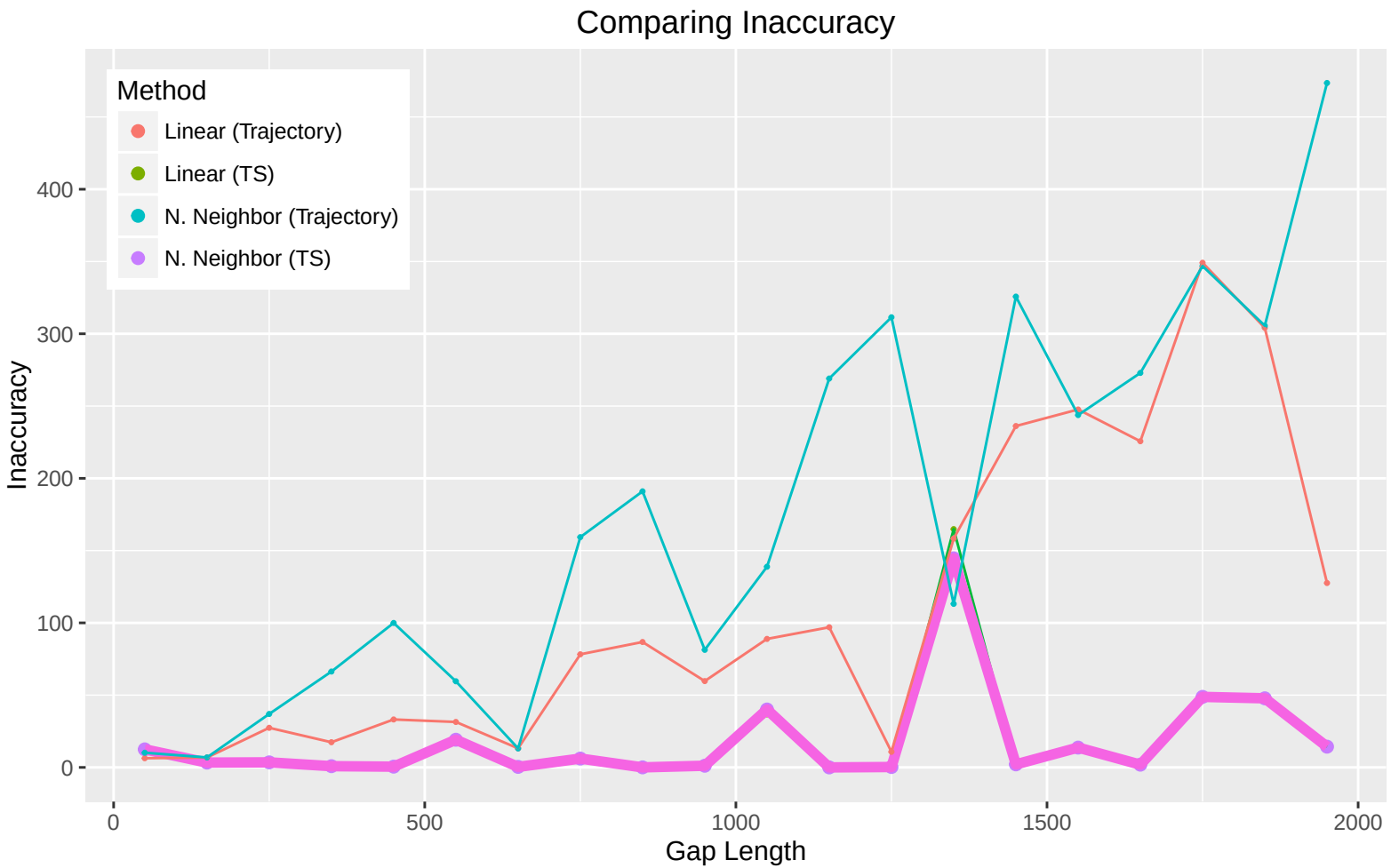
Combined Results: Inaccuracy



Combined Results: Inaccuracy



Combined Results: Inaccuracy



Statistical Difference

A two sample t-test returns the statistical significance of the results.

P values of two sample t-tests

		Time Series	
		Linear	N. Neighbor
Trajectory	Linear	0.001469728	0.001294356
	N. Neighbor	4.9942E-05	4.6472E-05

Additional Values (Linear vs Nearest Neighbor)

	P-value
Time Series	0.944246478
Trajectory	0.09547194

Observations

- Nearest neighbor and linear interpolation with time series return statistically identical results
- Time series interpolation categorically returned better results
- Nearest neighbor and linear interpolation with trajectory data also returned statistically identical results (however p-value is smaller)
  - This is likely a result of the small sample size
- Linear interpolation with trajectory data performed better than the nearest neighbor interpolation

Observations

- Nearest neighbor and linear interpolation with time series return statistically identical results
- Time series interpolation categorically returned better results
- Nearest neighbor and linear interpolation with trajectory data also returned statistically identical results (however p-value is smaller)
  - This is likely a result of the small sample size
- Linear interpolation with trajectory data performed better than the nearest neighbor interpolation

Observations

- Nearest neighbor and linear interpolation with time series return statistically identical results
- Time series interpolation categorically returned better results
- Nearest neighbor and linear interpolation with trajectory data also returned statistically identical results (however p-value is smaller)
  - This is likely a result of the small sample size
- Linear interpolation with trajectory data performed better than the nearest neighbor interpolation

Observations

- Nearest neighbor and linear interpolation with time series return statistically identical results
- Time series interpolation categorically returned better results
- Nearest neighbor and linear interpolation with trajectory data also returned statistically identical results (however p-value is smaller)
  - This is likely a result of the small sample size
- Linear interpolation with trajectory data performed better than the nearest neighbor interpolation

Spline Interpolation

Calculation of a series of polynomial interpolants which together form a piecewise polynomial function

The "spline" is the resulting piecewise polynomial function that passes through each known point.

$Q = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

For each $(x_i, y_i), (x_{i+1}, y_{i+1}) \in Q$ and for $D = [x_i, x_{i+1}]$, $R = [y_i, y_{i+1}]$, the problem of spline interpolation is to find a polynomial of a predetermined degree $P_i : D \to R$.

The spline, $S(x)$ is defined as

$$S(x) = \begin{cases} P_1(x) & : x_1 \leq x \leq x_2 \\ P_2(x) & : x_2 < x \leq x_3 \\ \vdots & \vdots \\ P_{n-1}(x) & : x_{n-1} < x \leq x_n \end{cases}$$

Spline Interpolation

Calculation of a series of polynomial interpolants which together form a piecewise polynomial function

The "spline" is the resulting piecewise polynomial function that passes through each known point.

$Q = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

For each $(x_i, y_i), (x_{i+1}, y_{i+1}) \in Q$ and for $D = [x_i, x_{i+1}]$, $R = [y_i, y_{i+1}]$, the problem of spline interpolation is to find a polynomial of a predetermined degree $P_i : D \to R$.

The spline, $S(x)$ is defined as

$$S(x) = \begin{cases} P_1(x) & : x_1 \leq x \leq x_2 \\ P_2(x) & : x_2 < x \leq x_3 \\ \vdots & \vdots \\ P_{n-1}(x) & : x_{n-1} < x \leq x_n \end{cases}$$

Spline Interpolation

Calculation of a series of polynomial interpolants which together form a piecewise polynomial function

The "spline" is the resulting piecewise polynomial function that passes through each known point.

$Q = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

For each $(x_i, y_i), (x_{i+1}, y_{i+1}) \in Q$ and for $D = [x_i, x_{i+1}]$, $R = [y_i, y_{i+1}]$, the problem of spline interpolation is to find a polynomial of a predetermined degree $P_i : D \to R$.

The spline, $S(x)$ is defined as

$$S(x) = \begin{cases} P_1(x) & : x_1 \leq x \leq x_2 \\ P_2(x) & : x_2 < x \leq x_3 \\ \vdots & \vdots \\ P_{n-1}(x) & : x_{n-1} < x \leq x_n \end{cases}$$

Spline Interpolation

Calculation of a series of polynomial interpolants which together form a piecewise polynomial function

The "spline" is the resulting piecewise polynomial function that passes through each known point.

$Q = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

For each $(x_i, y_i), (x_{i+1}, y_{i+1}) \in Q$ and for $D = [x_i, x_{i+1}]$, $R = [y_i, y_{i+1}]$, the problem of spline interpolation is to find a polynomial of a predetermined degree $P_i : D \to R$.

The spline, $S(x)$ is defined as

$$S(x) = \begin{cases} P_1(x) & : x_1 \leq x \leq x_2 \\ P_2(x) & : x_2 < x \leq x_3 \\ \vdots & \vdots \\ P_{n-1}(x) & : x_{n-1} < x \leq x_n \end{cases}$$

Spline Interpolation

Calculation of a series of polynomial interpolants which together form a piecewise polynomial function

The "spline" is the resulting piecewise polynomial function that passes through each known point.

$Q = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

For each $(x_i, y_i), (x_{i+1}, y_{i+1}) \in Q$ and for $D = [x_i, x_{i+1}]$, $R = [y_i, y_{i+1}]$, the problem of spline interpolation is to find a polynomial of a predetermined degree $P_i : D \to R$.

The spline, $S(x)$ is defined as

$$S(x) = \begin{cases} P_1(x) & : x_1 \leq x \leq x_2 \\ P_2(x) & : x_2 < x \leq x_3 \\ \vdots & \vdots \\ P_{n-1}(x) & : x_{n-1} < x \leq x_n \end{cases}$$

Spline Interpolation

Calculation of a series of polynomial interpolants which together form a piecewise polynomial function

The "spline" is the resulting piecewise polynomial function that passes through each known point.

$Q = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

For each $(x_i, y_i), (x_{i+1}, y_{i+1}) \in Q$ and for $D = [x_i, x_{i+1}]$, $R = [y_i, y_{i+1}]$, the problem of spline interpolation is to find a polynomial of a predetermined degree $P_i : D \to R$.

The spline, $S(x)$ is defined as

$$S(x) = \begin{cases} P_1(x) & : x_1 \leq x \leq x_2 \\ P_2(x) & : x_2 < x \leq x_3 \\ \vdots & \vdots \\ P_{n-1}(x) & : x_{n-1} < x \leq x_n \end{cases}$$

Spline Interpolation

Linear interpolation is an instance of spline interpolation where $\deg(P_i) = 1$.

Cubic spline interpolation is also widely used.

Definition

$$\deg(P_i) = 3$$

$$P_i'(x_i) = P_i'(x_{i+1})$$

$$P_i''(x_i) = P_i''(x_{i+1})$$

Future work with this.

Curve fitting and Polynomial Interpolation

Definition

Given a set of n points, find a polynomial of degree $n-1$ that passes through each point.

Important properties:

- The polynomial can then be used to interpolate any desired value
- Global interpolant
- Yields smoother interpolants than those generated through nearest neighbor, linear, or cubic spline means
- Runge's phenomenon can cause issues

Curve fitting and Polynomial Interpolation

Definition

Given a set of n points, find a polynomial of degree $n-1$ that passes through each point.

Important properties:

- The polynomial can then be used to interpolate any desired value
- Global interpolant
- Yields smoother interpolants than those generated through nearest neighbor, linear, or cubic spline means
- Runge's phenomenon can cause issues

Introduction to Velocity-Based Model

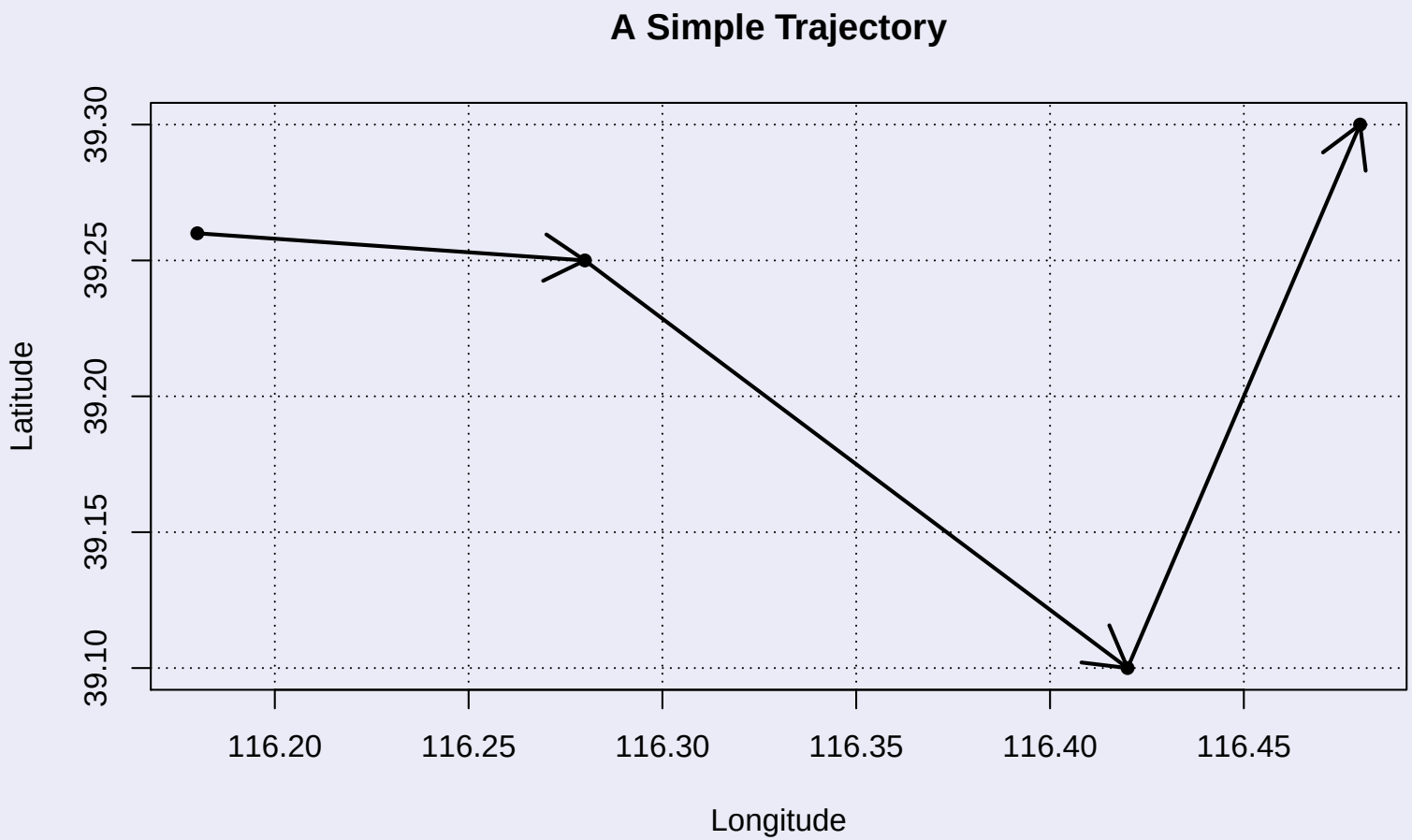
Limitations of nearest neighbor and linear interpolation methods

- Highly innaccurate
- Information is "wasted"
- Nearest neighbor relies on exact interpolation
- Interpolated paths are highly inflexible
- Both are local interpolation methods

Minimizing these limitations requires a model that extracts information from known trajectory points.

Motivation

The trajectory can be viewed as a set of vectors



Motivation

Properties of vectors

- Direction
- Magnitude

Motivation

Properties of vectors

- Direction
- Magnitude

Properties of spatial vectors

- Compass direction
- Velocity

We can exploit these properties to yield more accurate interpolants.

Function Outline

Our algorithm consists of three main steps

1. Parse known values and create a dataset
2. Interpolate
    1. Find a matching data point
    2. Extract information based on this match
    3. Apply information to calculation of interpolated point
3. Update dataset

Function Outline

Our algorithm consists of three main steps

1. Parse known values and create a dataset
2. Interpolate
    1. Find a matching data point
    2. Extract information based on this match
    3. Apply information to calculation of interpolated point
3. Update dataset

Function Outline

Our algorithm consists of three main steps

1. Parse known values and create a dataset
2. Interpolate
    1. Find a matching data point
    2. Extract information based on this match
    3. Apply information to calculation of interpolated point
3. Update dataset

Parsing

Each point has five defining values:

1. Time of day
2. Latitude
3. Longitude
4. **Change in direction**
5. **Velocity**

The latter two quantities are calculated for each point (with the exception of the last point).

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Interpolation and Update

Interpolation begins at the leftmost end of the leftmost gap.

Velocity of the closest known point (nearest neighbor) is recorded.

Selection routine returns a set of potential matches.

- Searches dataset for points whose velocities are within a user-defined threshold of given velocity
- If there are not any values, the routine returns the point with the closest velocity

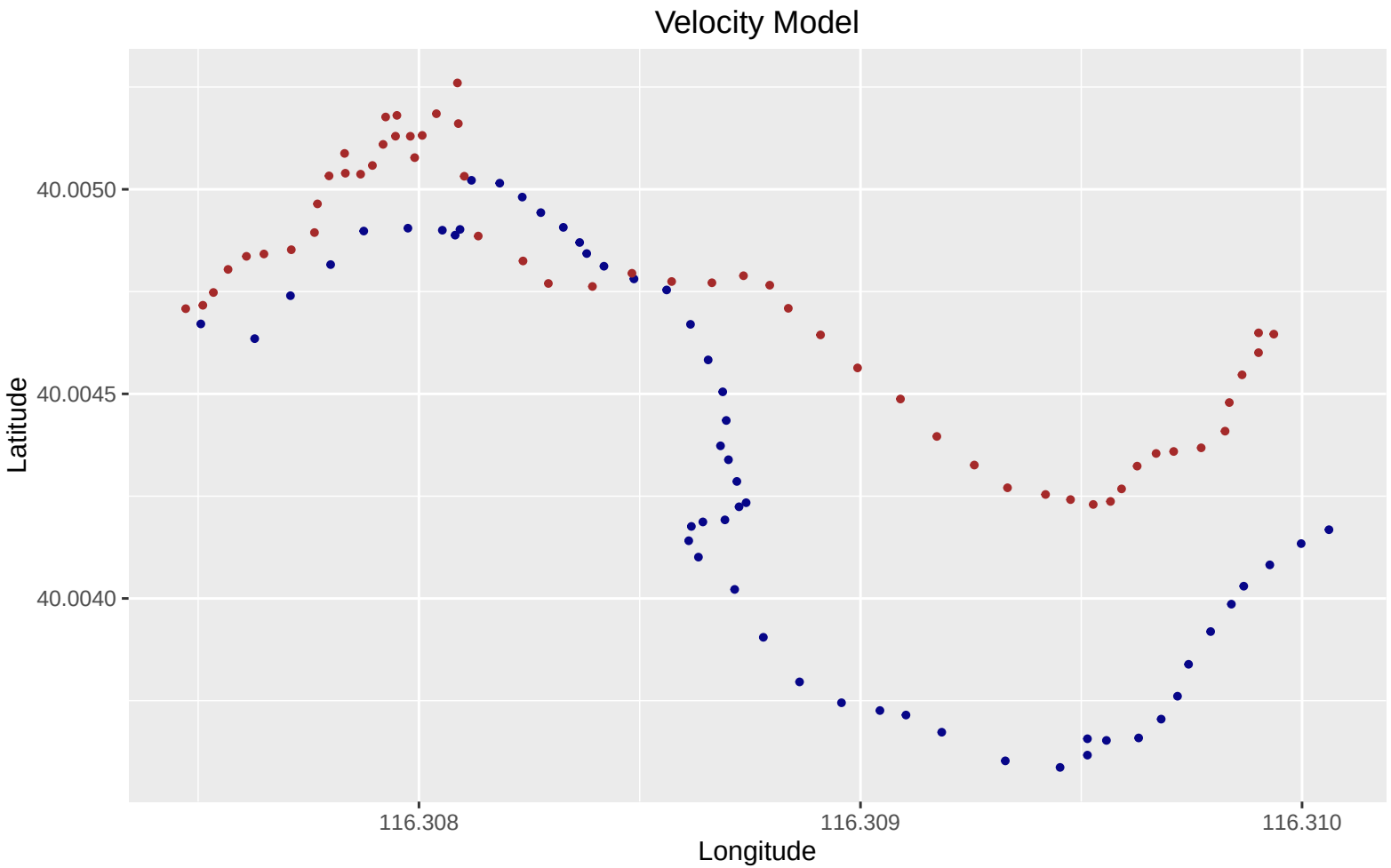
Time of day serves as a second metric (the point with the closest time value is used).

The direction and acceleration of this point is recorded.

These values are applied to the velocity and position of the nearest neighbor to yield an interpolated point (time is incremented by five seconds).

The values of the new point are incorporated into the dataset of known values.

Initial Testing



Future Work

Considerations for furutre research

- Automatic parameter estimation and model refinement
- Cubic spline interpolation
- Polynomial curve fitting
- Probabilistic modeling
- Adaptive interpolation method (combining models)

Future Work

Considerations for furutre research

- Automatic parameter estimation and model refinement
- Cubic spline interpolation
- Polynomial curve fitting
- Probabilistic modeling
- Adaptive interpolation method (combining models)

Future Work

Considerations for furutre research

- Automatic parameter estimation and model refinement
- Cubic spline interpolation
- Polynomial curve fitting
- Probabilistic modeling
- Adaptive interpolation method (combining models)

Future Work

Considerations for furutre research

- Automatic parameter estimation and model refinement
- Cubic spline interpolation
- Polynomial curve fitting
- Probabilistic modeling
- Adaptive interpolation method (combining models)

Future Work

Considerations for furutre research

- Automatic parameter estimation and model refinement
- Cubic spline interpolation
- Polynomial curve fitting
- Probabilistic modeling
- Adaptive interpolation method (combining models)