Linking As-Planned and As-Executed Machine Data in Near Real Time

Michael Roa, michael.roa2@gmail.com

University of Maryland
NIST/Engineering Laboratory/
Systems Integration Division

Overview

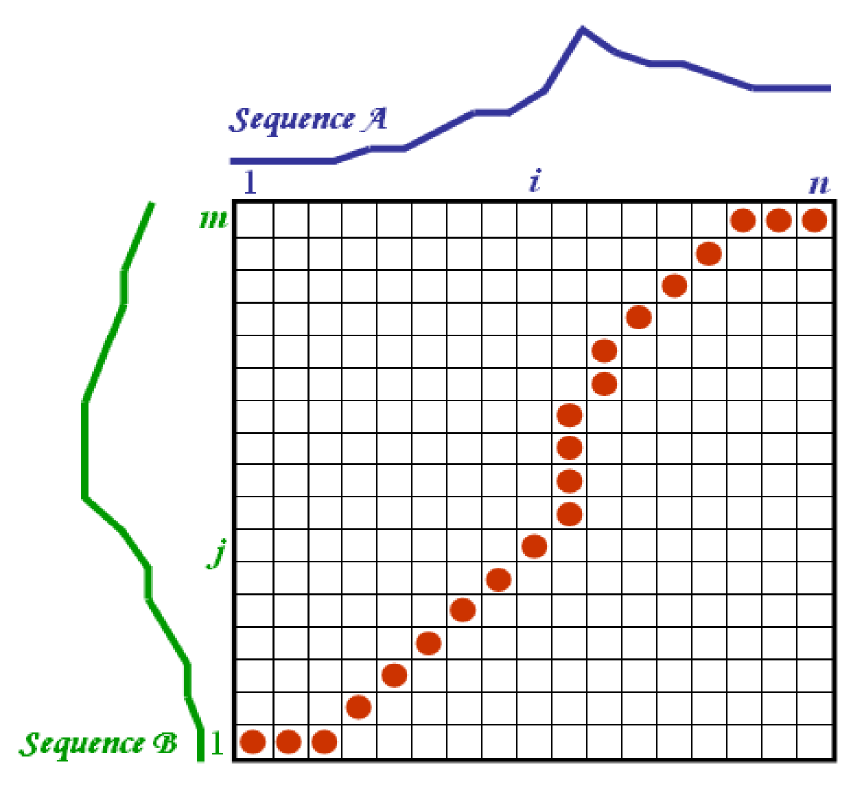
- Background
- Dynamic Time Warping
- Investigating Near Real Time Solutions
- Windowed Time Warping
- Implementation
- Results
- Conclusion

Motivation

- Data mapping as-planned and as-executed data improves manufacturing
- Tell us machines behavior
- Helps optimize product manufacturing, maintain machines and more
- Analyzing data faster vastly improves decision making
- Allows for more planning and flexibility to take on challenges

Dynamic Time Warping (DTW)

- Current method of data linking is Dynamic Time Warping (DTW)
- Links two similar data sequences
- Identifies discrepancies between the two
- Creates a merged data set
- Currently implemented in R



DTW Common Algorithm

Typical DTW algorithm

Computes distance cost of each element

Lowest distance becomes return sequence

Complexity is determined by the size of input arrays, $O(n^2)$

$$D(i,j) = |t(i) - r(j)| + \min\begin{cases} D(i-1,j) \\ D(i-1,j-1) \\ D(i,j-1) \end{cases}$$

Faults of DTW

- Slow
- Not standalone
- Requires complete data set
- Not capable of near real time
- Current implementation in R cannot accomplish near real time DTW

s=5=n (n,m)

3	17	7	9	9	10	9
5	14	7	6	11	7	11
2	9	5	8	6	10	11
5	7	4	4	9	10	14
2	2	3	7	9	13	14
0	0	3	6	0	6	1

t=6=m

(0,0)

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
   DTW := array [0..n, 0..m]

   for i := 1 to n
       DTW[i, 0] := infinity
   for i := 1 to m
       DTW[0, i] := infinity
   DTW[0, 0] := 0

   for i := 1 to n
       for j := 1 to m
           cost := d(s[i], t[j])
           DTW[i, j] := cost + minimum(DTW[i-1, j  ],
                                       DTW[i  , j-1],
                                       DTW[i-1, j-1])

   return DTW[n, m]
}
```

Introducing Near Real Time DTW

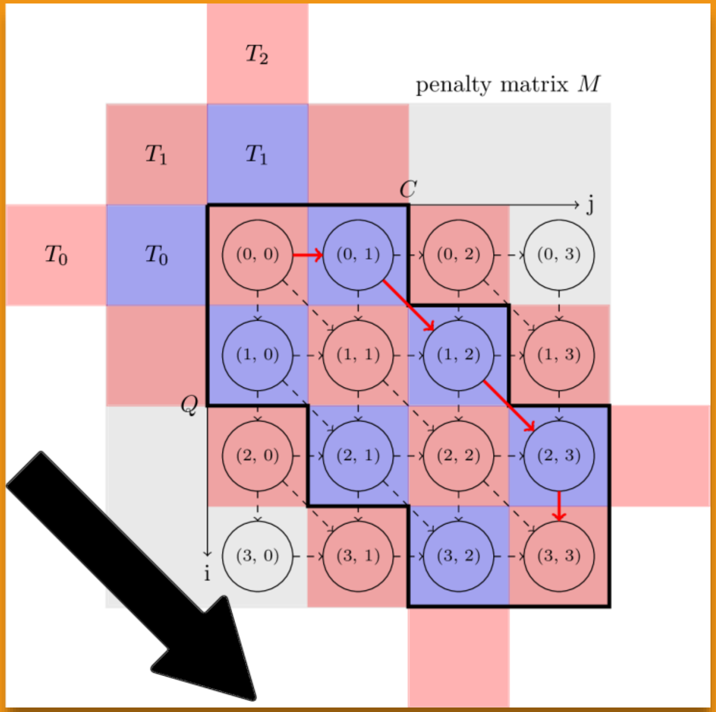
Near real time solutions will allow for effective decision making

Requirements:

- Faster than R implementation
- Similarly accurate to DTW results
- Can intake data in near real time
- Standalone application

Investigating Near Real Time DTW Methods

- Multiple methods and techniques
- Pros and cons to each
  - Traditional DTW; slow and memory hungry, $O(n^2)$
  - FastDTW; memory efficient, full data set, $O(n)$
  - Windowed Time Warping; short-term, fast, $O(n)$
- Windowed Time Warping and FastDTW offered the best advantages for the goals of project



Effectiveness of Fast Dynamic Time Warping (FastDTW)

- Calculates the path at a lower resolution for an estimate
- Next pathing uses estimate as basis
- Accurate and quick
- Sakoe-Chiba band to cull un-necessary calculations
- However, FastDTW does not fit the project goals
- FastDTW requires a full dataset. This would not be near real time

Effectiveness of Windowed Time Warping (WTW)

- Uses A-Star algorithm at lower resolutions
- “Hops” window along sections of matrix
- Accurate and quick
- Does not calculate outside of windows/bounds
- Assumedly able to work with an incomplete data set.
- Best suited for near real time

Windowed Time Warping Algorithm

Significant differences from the common DTW approaches.

Complexity is $O(n)$

Accuracy can rival matches made through DTW approaches

Extremely fast compared to DTW

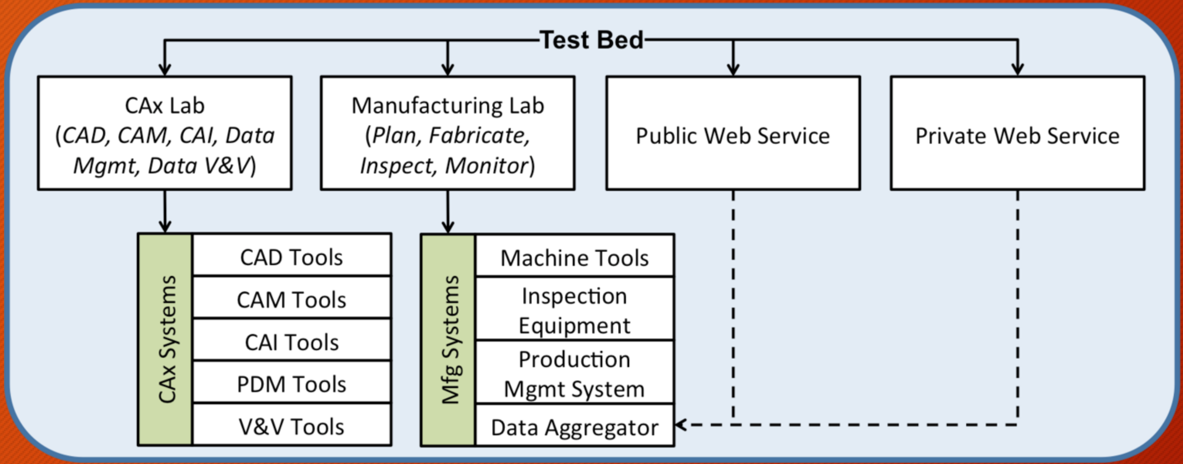
Can utilize smaller frames of data, yet still achieve good accuracy

```
Input: Feature Sequence A and Feature Sequence B
Output: Alignment Path
Path = new Path.starting(1,1);
while Path.length < min (A.length,B.length) do
    Start = Path.end;
    End = Start;
    while (End - Start).length < Window_Size do
        End =
        argmin(Inner_Product(End.next_points));
    end
    Cost_Estimate = End.cost;
    A-Star_Matrix =
    A_Star_Fill_Rect(Start,End,Cost_Estimate);
    Path.add(A_Star_Matrix.getPath(1,Hop_Size));
end
return Path;
```


```
Input: Feature Sequence A and Feature Sequence B
Output: Alignment Path
Path = new Path.starting(1,1);
while Path.length < min (A.length,B.length) do
    Start = Path.end;
    End = Start;
    while (End - Start).length < Window_Size do
        End =
        argmin(Inner_Product(End.next_points));
    end
    Cost_Estimate = End.cost;
    A-Star_Matrix =
    A_Star_Fill_Rect(Start,End,Cost_Estimate);
    Path.add(A_Star_Matrix.getPath(1,Hop_Size));
end
return Path;
```

Smart Manufacturing Systems Test Bed

- Manufacturing floor with Computer Numerical Controlled(CNC) machines
- Machines execute G-Code, report back MTConnect data
- G-Code is pre-planned data, MTConnect is as-executed data
- This is the main data source for testing the application



Testing

- Using different pre-executed data sets collected from SMS Test Bed
- Measured speed of program, speed of algorithm, accuracy
- Time key function execution with system timers
- Time DTW function
- Compare the ‘true’ DTW pathing to the computed WTW pathing

Results - Speed

R Code Execution Time	Trial 1	Trial 2	Trial 3	Avg	Java Code Execution Time	Trial 1	Trial 2	Trial 3	Avg
Gcode Parser	13.92	14.15	14.67	14.24		0.0024	0.0014	0.0016	0.0018
Gcode Simulator	7.12	6.88	6.85	6.95		0.0244	0.0225	0.0230	0.0233
MTConnect Parser	13.67	12.67	13.28	13.20		0.0328	0.0358	0.0310	0.0332
DTW Simulation	6.31	6.4	6.35	6.35		5.55E-05	4.59E-05	5.1E-05	5.06667E-05
WTW Simulation						2.51E-06	1.84E-06	2.9E-06	2.41333E-06

$$\frac{|v_1 - v_2|}{\frac{(v_1 + v_2)}{2}} \times 100 = \text{Percent Change}$$

100% decrease versus R (Avg. execution time percent difference)
95% decrease than DTW (Avg. execution time percent difference)

Results - Accuracy

Correct Data Pathings			WTW Shared Pathings		
Window Set 1	Window Set 2	Window Set 3	Window Set 1	Window Set 2	Window Set 3
20	40	100	16	32	81
21	42	63	13	27	43
7	17	22	7	15	17
			75%	74%	76%

$$\frac{WTW\ Average}{DTW\ Average} * 100 = \text{Percent Accuracy}$$

DTW gives 100% correct pathing of the data.
Same answers as are considered accurate.
≈75% pathing accuracy (Avg. percent pathing shared)

Conclusion

- Windowed Time Warping shows promise for near real time
- Real time data testing in progress
- Best mix of hop size, window size?
- Introduce constraints to restrict calculations

Thank you!

Michael Roa, michael.roa2@gmail.com

Advisor: Moneer Helu

University of Maryland

NIST, Systems Integration Division