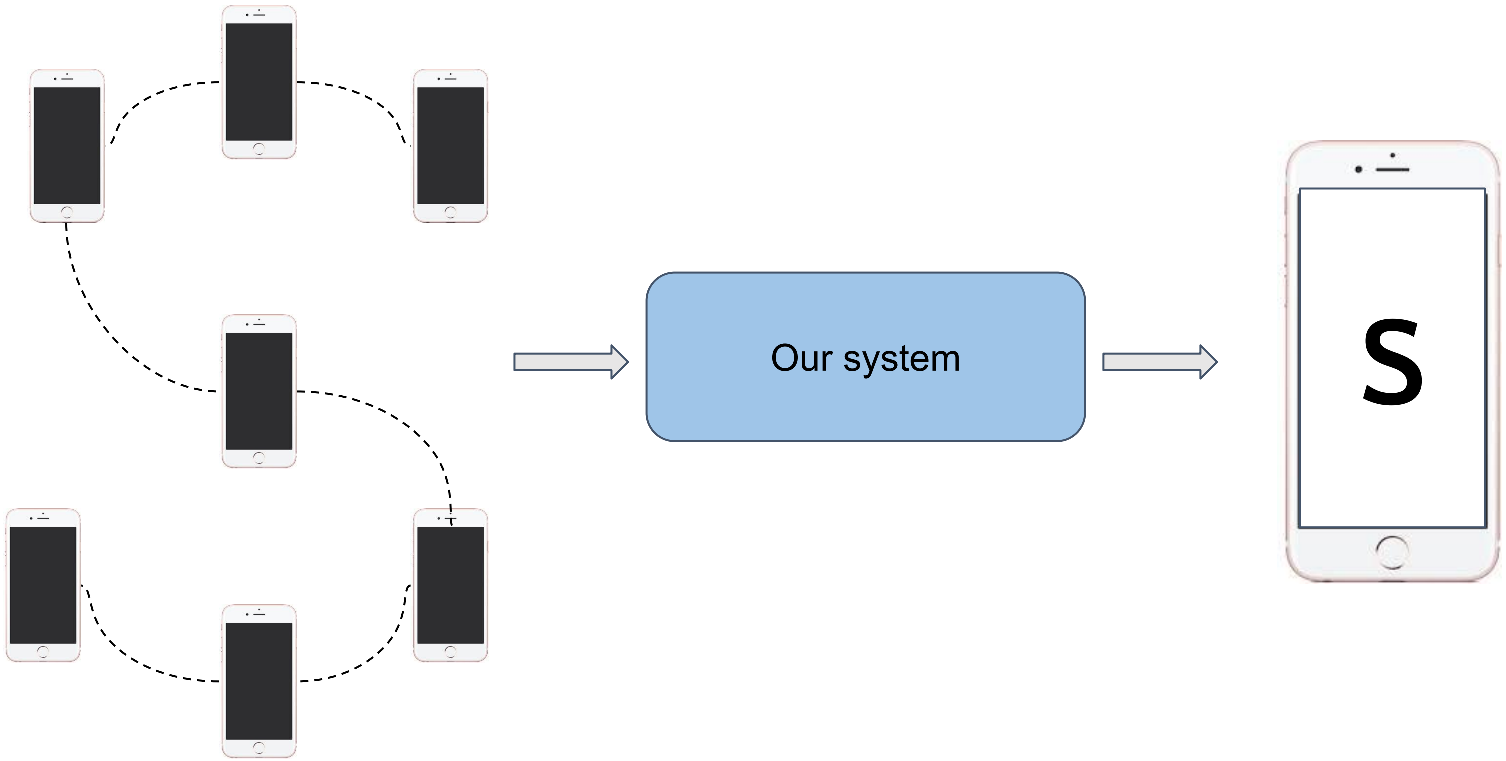
Problem Statement

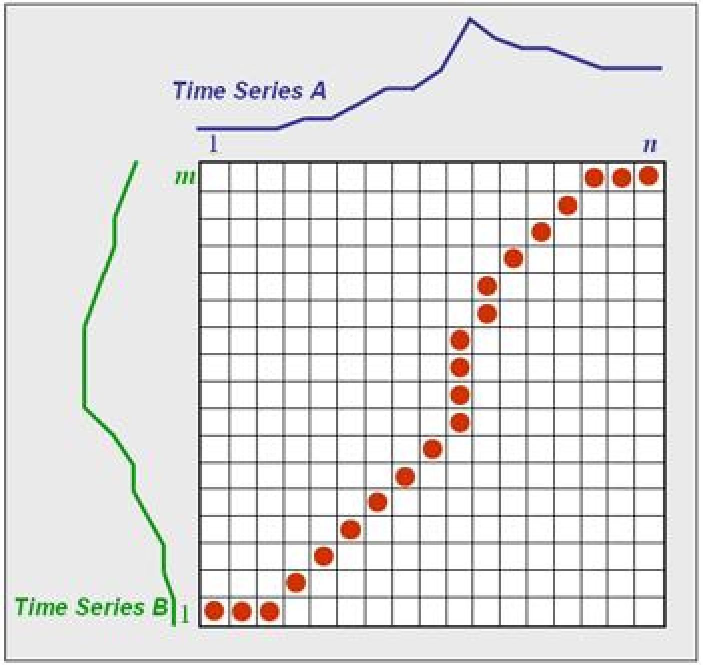


Dynamic Time Warping (DTW)

Objective: measure similarities between two temporal sequences.

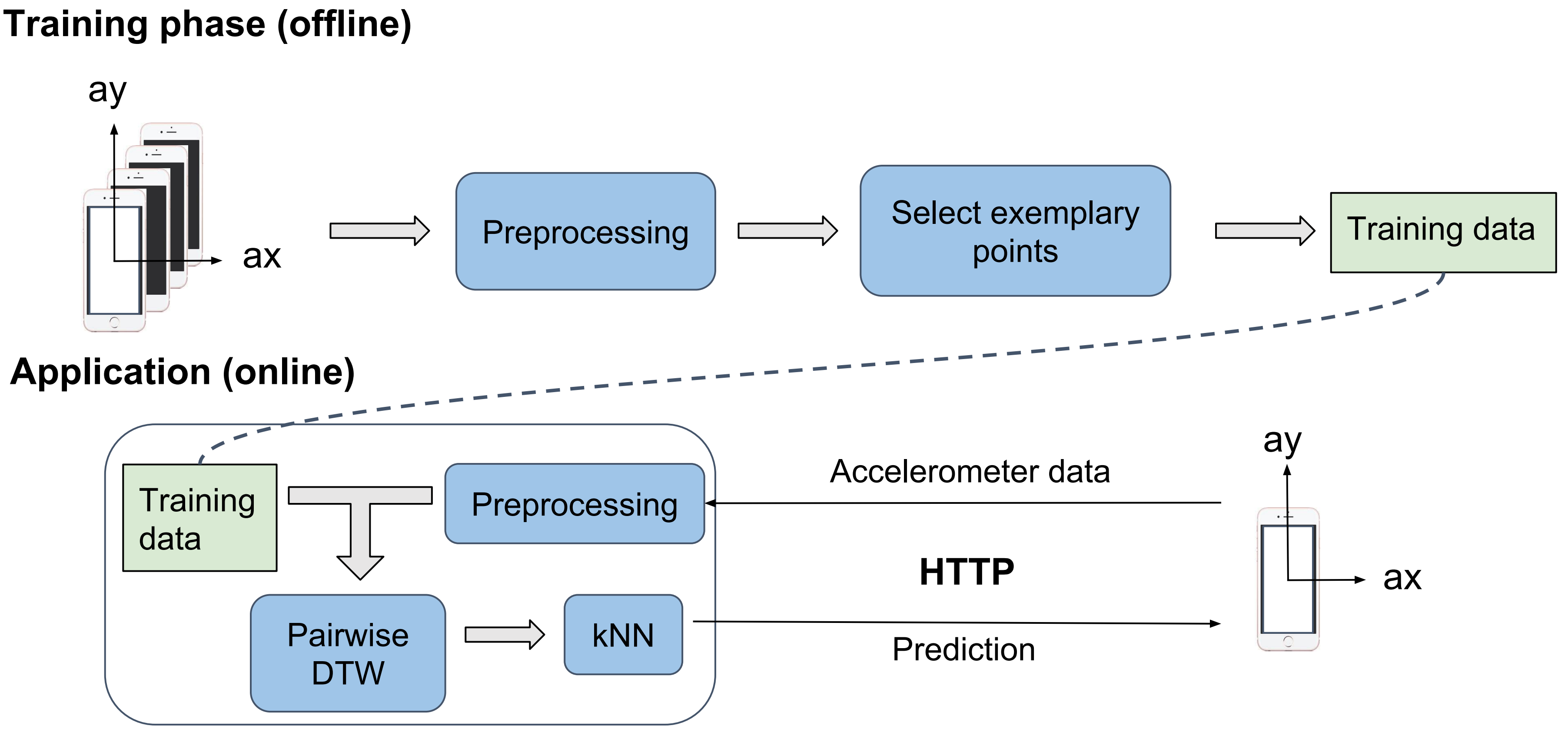
Idea: Given two sequences, warp them non-linearly in the time dimension, and calculate an optimal match between them.

Application: speech recognition, signature recognition, shape matching, etc.



(http://cst.tu-plovdiv.bg/bi/DTWimpute/DTWalgorithm.html)

System Infrastructure



Data Collection

General Setting:

- Letters: O ,I, J, L, Z, S, V, T, X, B.
- 10 people, each person writes each letter 10 times.
- Collection frequency: 100Hz

Constraints

1. Vertically hold, face towards the phone
2. Touch the screen, keep it touched until the end
3. Try to keep the phone vertically straight
4. Try to write the letter on the same plane as the phone’s plane

How to draw the letters

1. For B, go down and up and the bumps
2. For I, up -> down end.
3. For X, Start at the left top corner

Data Preprocessing, Distance Metric

Data Preprocessing

- Scaling: Scale ax and ay independently to be inside [-1, 1].
- Sampling: take the average over every n data points.

Distance Metric: For two data points (ax1, ay1) and (ax2, ay2),

$$d = \sqrt{(a_{x1} - a_{x2})^2 + (a_{y1} - a_{y2})^2}$$

Exemplary Points for kNN

For each user’s data, and each letter,

1. Calculate sum{DTW distances from all other time series} for each time series.
2. **Remove outliers**
   - Calculate the mean and standard deviation of the sums
   - Remove time series *i* if sum_i is 2 standard deviation away from the mean
3. **Choose Exemplary points**

   Select *n* time series with smallest sum of distances from other time series.

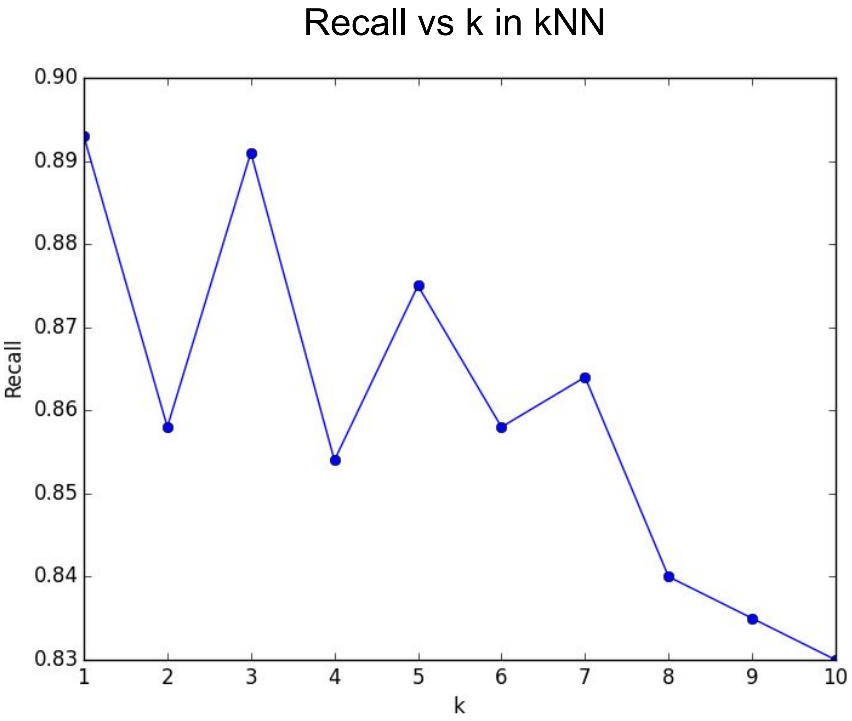
Evaluation

- Train on 9 people’s data, and test on the remaining person.
- Recall = number of correct predictions / number of air gestures.
- Results when
  - Sample every 10 data points,
  - Number of exemplaries per person per letter = 1
  - Number of Nearest Neighbors (k in kNN) = 1

user1	user2	user3	user4	user5	user6	user7	user8	user9	user10	Average
57%	79%	96%	96%	96%	83%	89%	96%	92%	94%	**88%**

User1 behaves differently from other users.

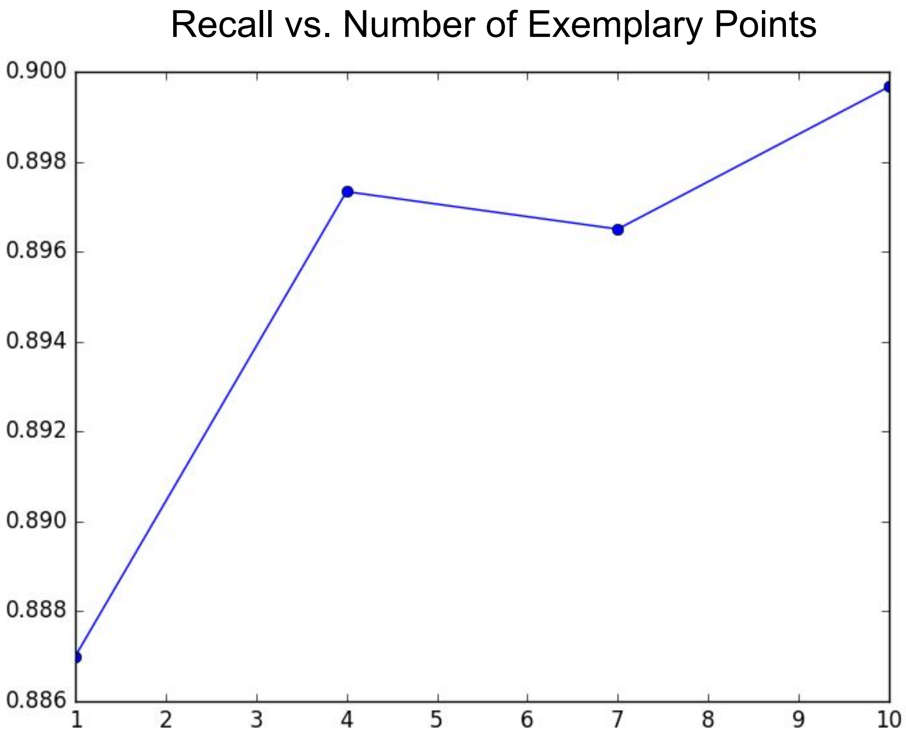
Recall vs. Number of Nearest Neighbors (k in kNN)



Number of exemplary points per person per letter = 1.
Sample over every 5 letter.

- In general, performance decreases when k increases.
- We choose **k = 1** in our system.

Recall vs. Number of Exemplary Points



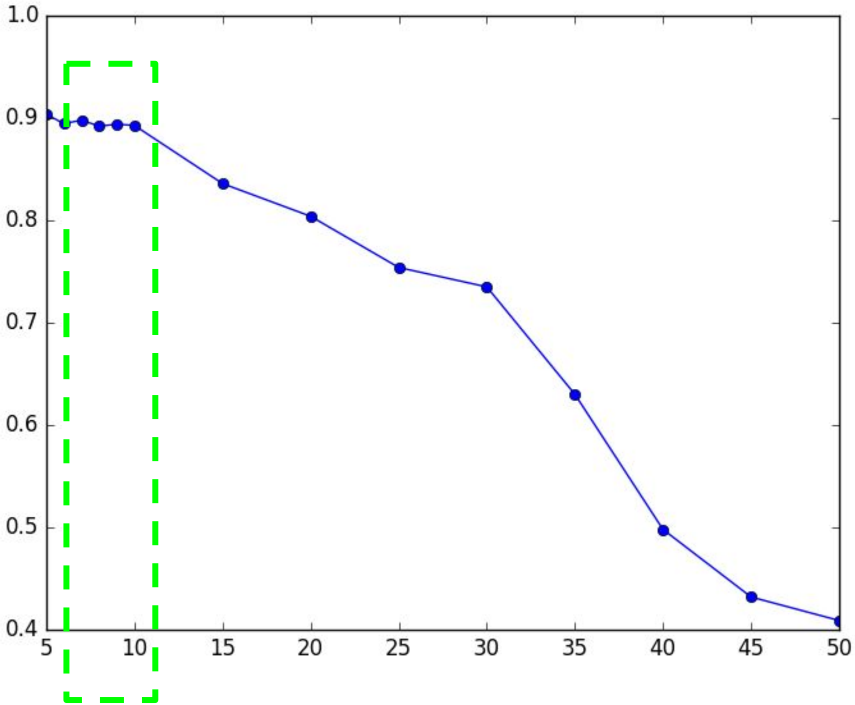
- But, runtime increase proportional to the number of exemplary points

- 1.4% increase vs. 10 times slower

  We choose **k = 1**

Recall vs. Sample Rate

Recall vs. Sample Rate



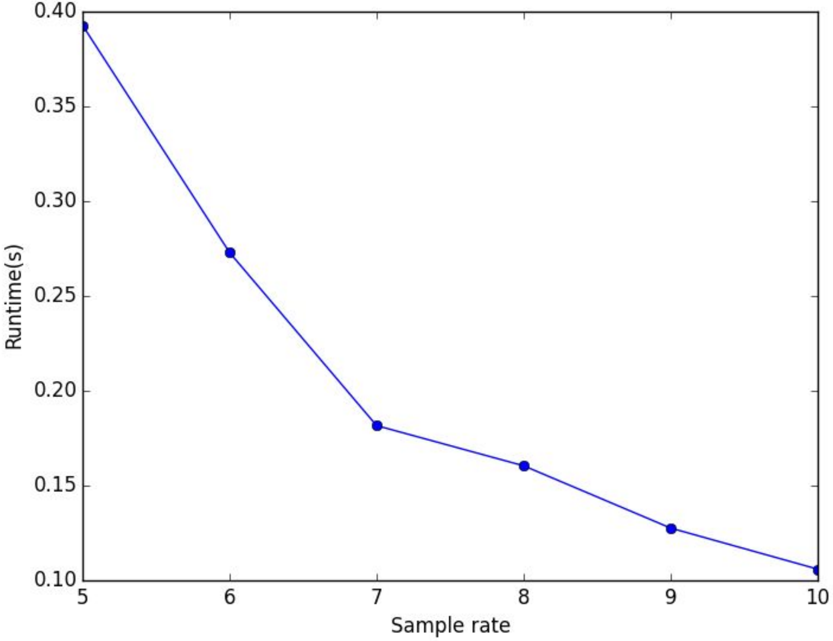
We choose

sample rate of 10

(See runtime comparisons)

Running Time vs. Sample Rate

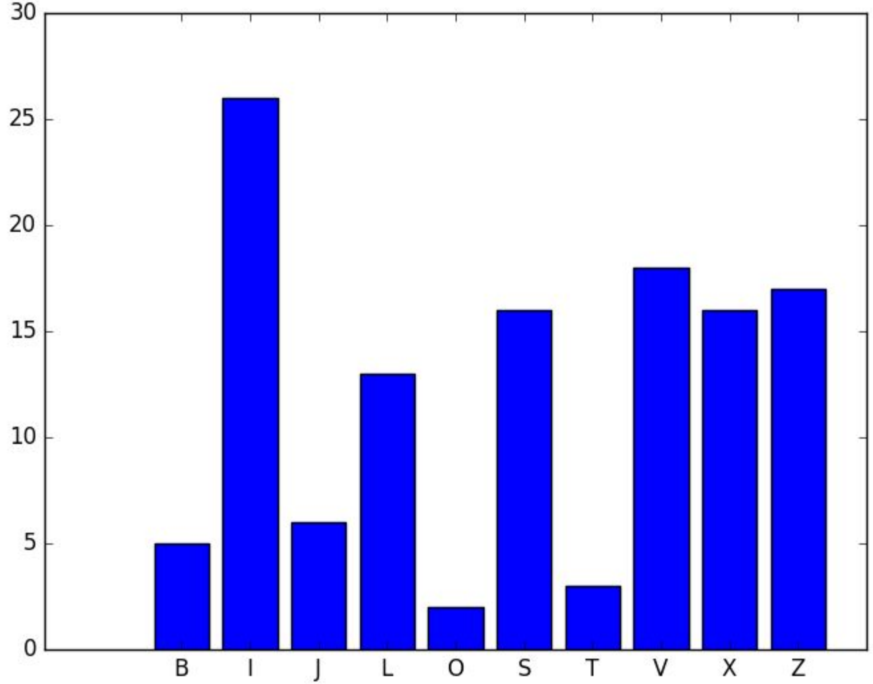
Runtime per 100 DTWs
vs sample rate



In order to decrease the end-to-end latency, we choose

sample rate = 10

Misclassification rate per letter

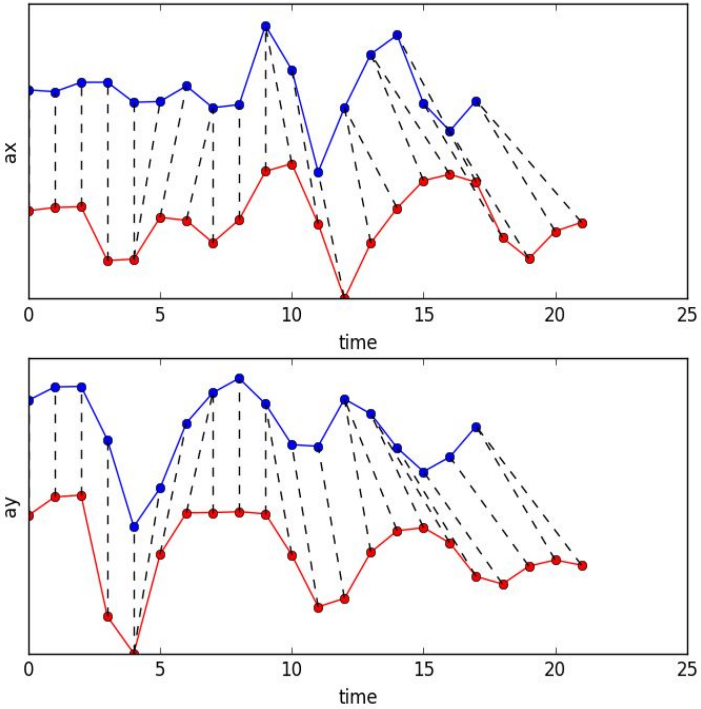


Parameters:

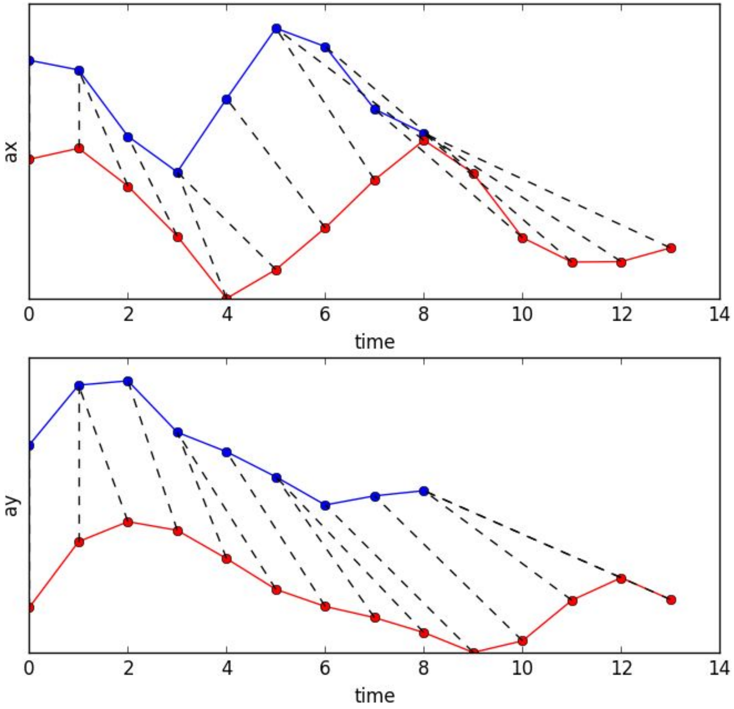
- Sample rate = 10
- Number of exemplary points = 1
- K for kNN = 1

Sequence Alignments

B



S


ax
time
ay
time