Time Dependent Data Mining

RAVEN Workshop

www.inl.gov

Idaho National Laboratory

Getting on the same page

- This workshop will include interactive examples
- paths given starting at raven (`raven/`)

```
~> cd projects/raven
~/projects/raven> git checkout workshop_2018
~/projects/raven> git pull
```

Slides are provided as well:

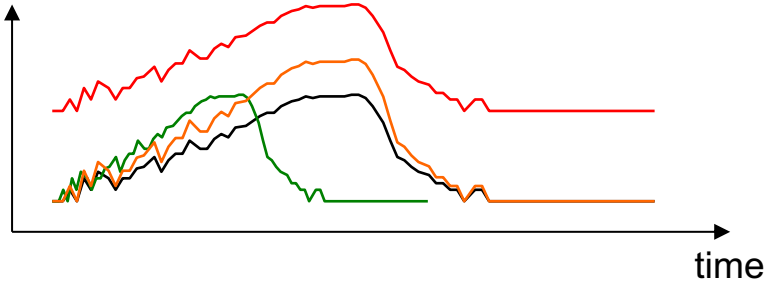
```
raven/doc/workshop/timeDepDataAnalysis
```

Overview

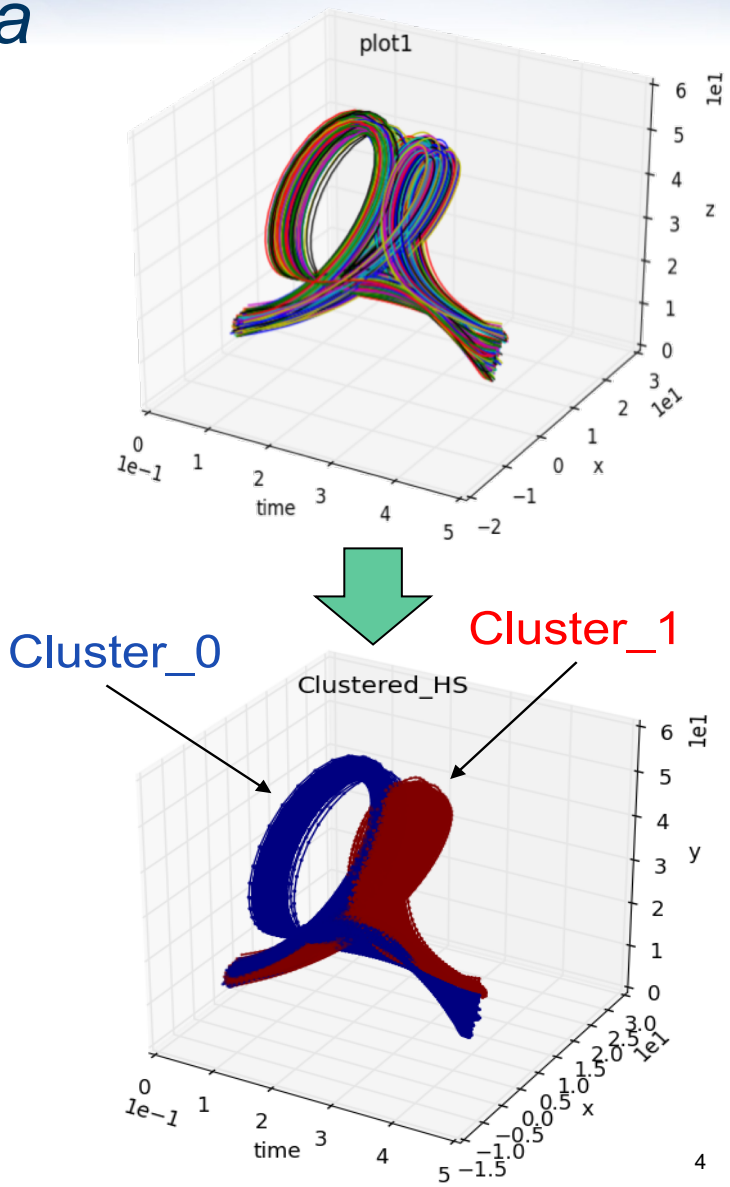
- Time-series post-processors
- Time dependent basic statistics
- Time slice clustering
- Time dependent clustering
  - Approach 1: time series transformation (K-Means)
  - Approach 2: time dependent metrics (Hierarchical)
    - Euclidean
    - Dynamic Time Warping

Clustering: Time-Dependent Data

- Objective: analyze time-dependent data
- Similarity can be subjective



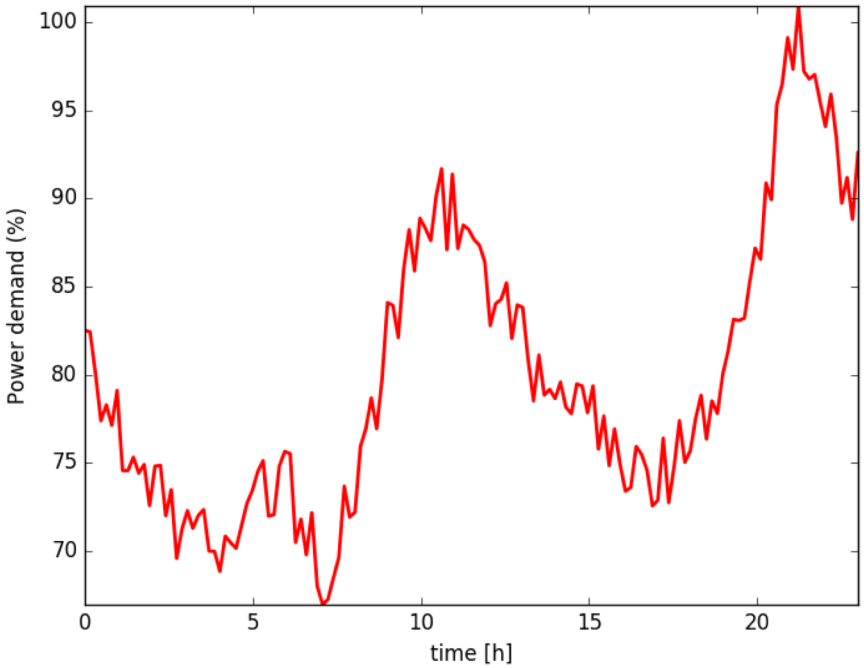
- Challenges
  - Different time lengths
  - Different sample rates
  - Presence of noise or missing data
  - Time delays

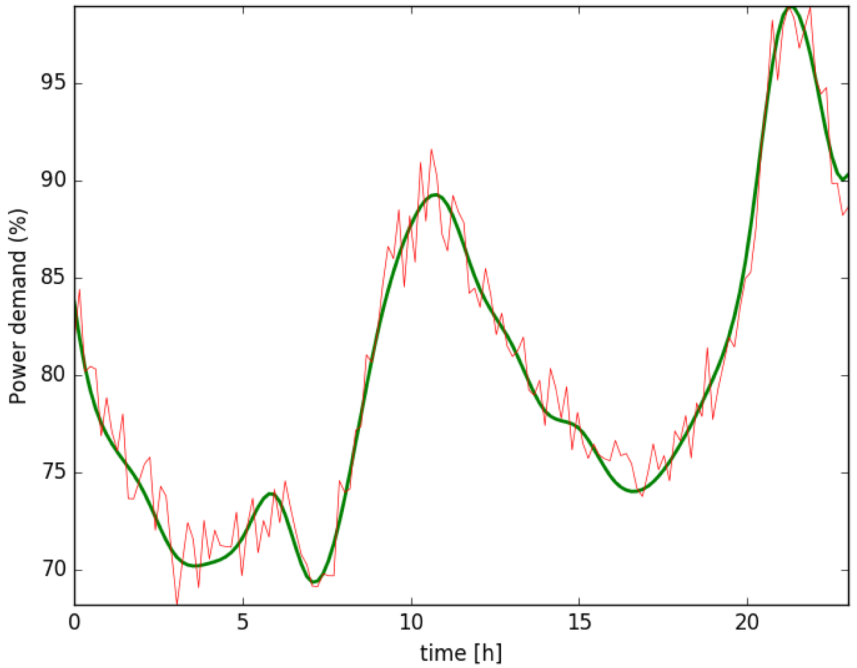


Data Pre-Processing: Smoothing

- Data filtering (e.g., KDE regression)

$$Q' = smooth(Q)$$

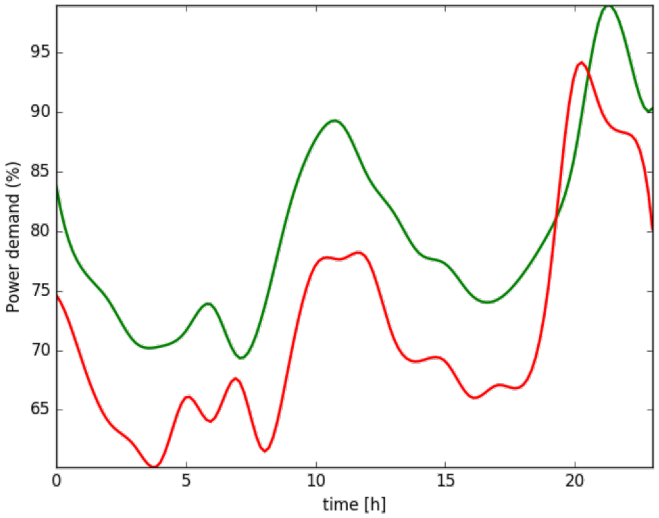




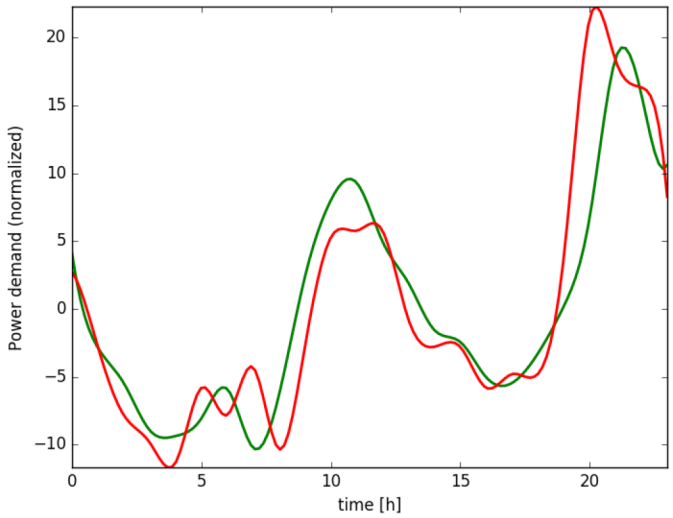
Data Source: *The UCR Time Series Classification Archive*, www.cs.ucr.edu/~eamonn/time_series_data/

Data Pre-Processing: Normalization

Offset Translation

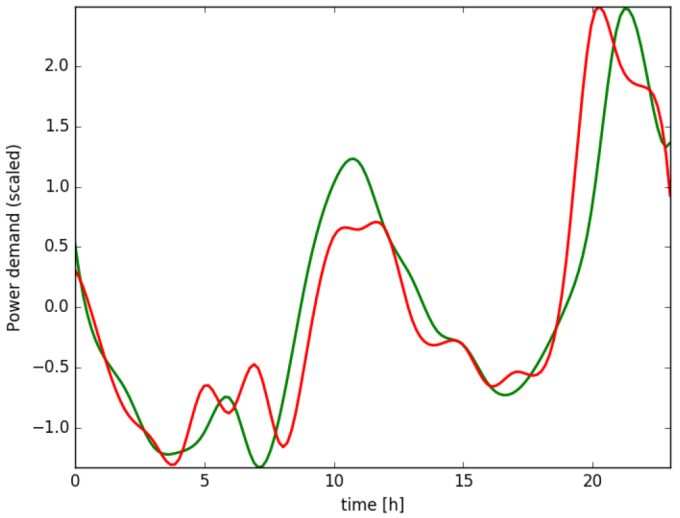


$$Q' = Q - mean(Q)$$



$$Q' = \frac{Q - mean(Q)}{std(Q)}$$

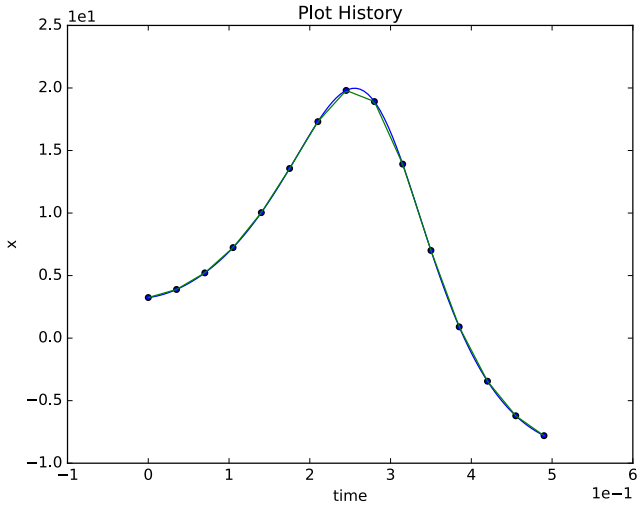
Amplitude Scaling



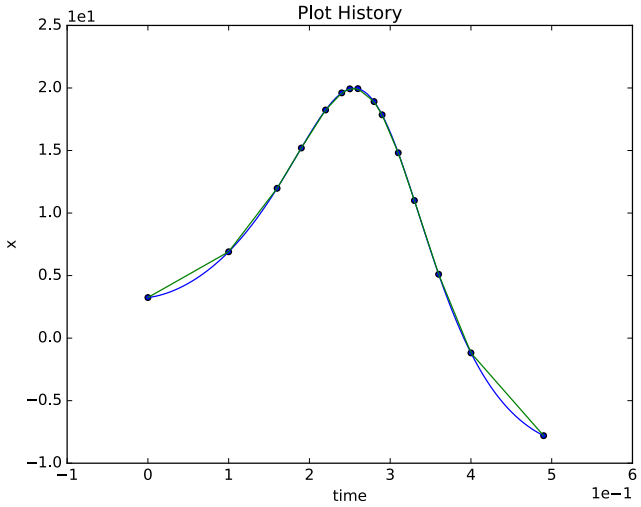
Data Pre-Processing: Re-Sampling

- Objective: reduce memory space of each time series

- Method: re-sampling the time series
  - Smartly locate sample points on strategically important regions
    - e.g. high derivative (gradient) regions



Uniform sampling



First-order derivative sampling

RAVEN Time-Series Post-Processors

- Class: Interfaced Post-Processors
  - RAVEN provides a generic interface to create user-defined generic Post-Processors
  - Act on both PointSets and HistorySets

RAVEN Time-Series Post-Processors: Examples

- HSPS: it converts an HistorySet into a PointSet
  - Each history is converted into a multi-dimensional vector
- HistorySetSampling
  - Original HistorySet is re-sampled accordingly to a specific sampling strategy
- HistorySetSync
  - Time series contained in the original HistorySet are synchronized in time
    - Identical initial and final time
    - Identical number of samples
- dataObjectLabelFilter
  - Filter the dataObject for a specific value of the clustering label

RAVEN Example 1
Time Dependent Basic Statistics

RAVEN Example 1: Time-Dep. Basic Statistics

- Steps
  1. Generate time-dependent data
  2. Post-Process the data
  3. Create a dataObject (PointSet) from processed data

RAVEN Example 1: Time-Dep. Basic Statistics

Distributions	Models	Samplers	Databases	DataObjects	Steps

```
<Models>
 ...
 <PostProcessor name="timeDepBasicStat" subType="BasicStatistics">
   <pivotParameter>time</pivotParameter>
   <pivotParameter>time</pivotParameter>
   <expectedValue prefix="mean" >x,y,z</expectedValue>
   <percentile prefix="percentile">x,y,z</percentile>
 </PostProcessor>
 ...
</Models>
```

Input file name: time_dep.xml

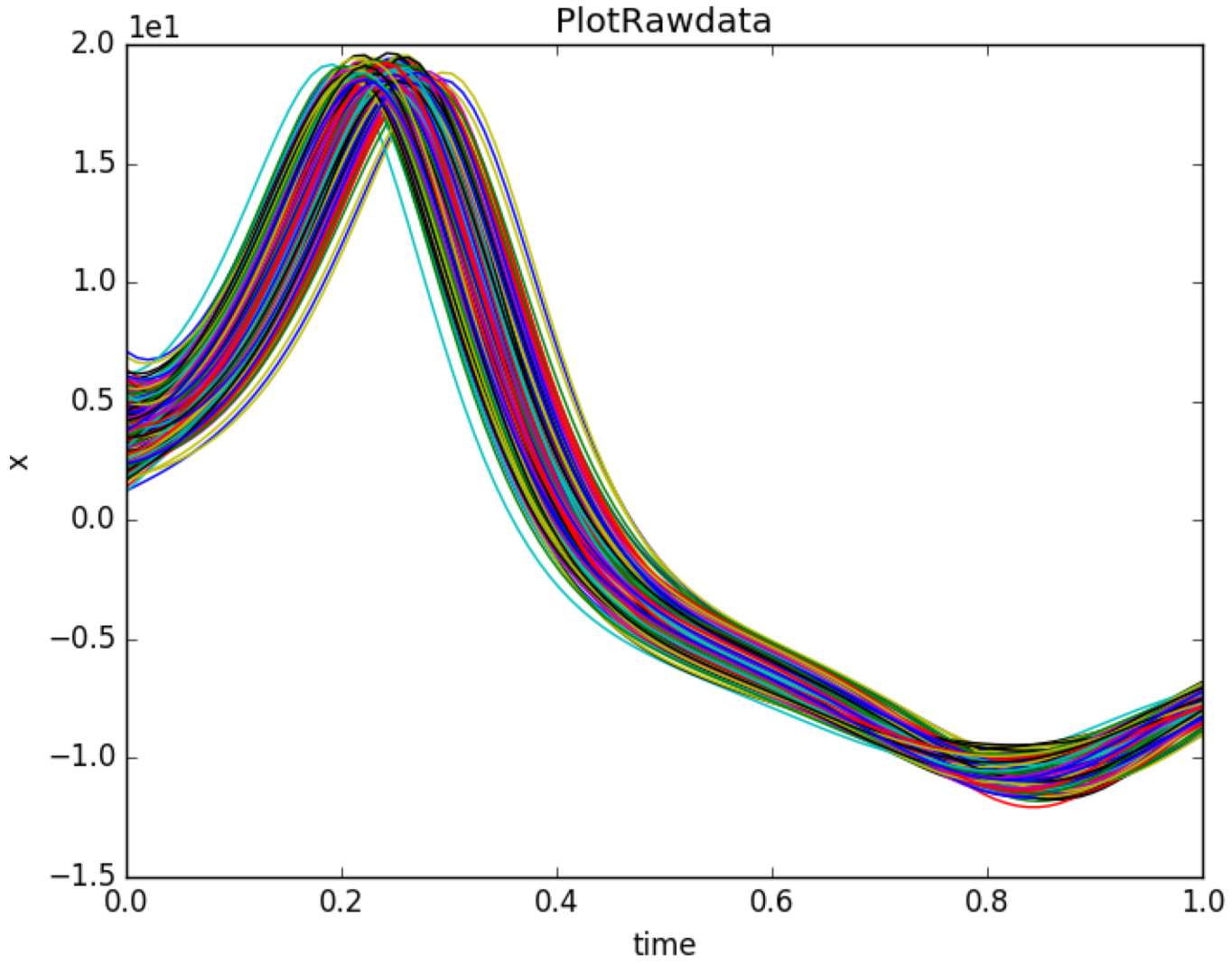
RAVEN Example 1: Time-Dep. Basic Statistics

Distributions	Models	Samplers	Databases	DataObjects	Steps

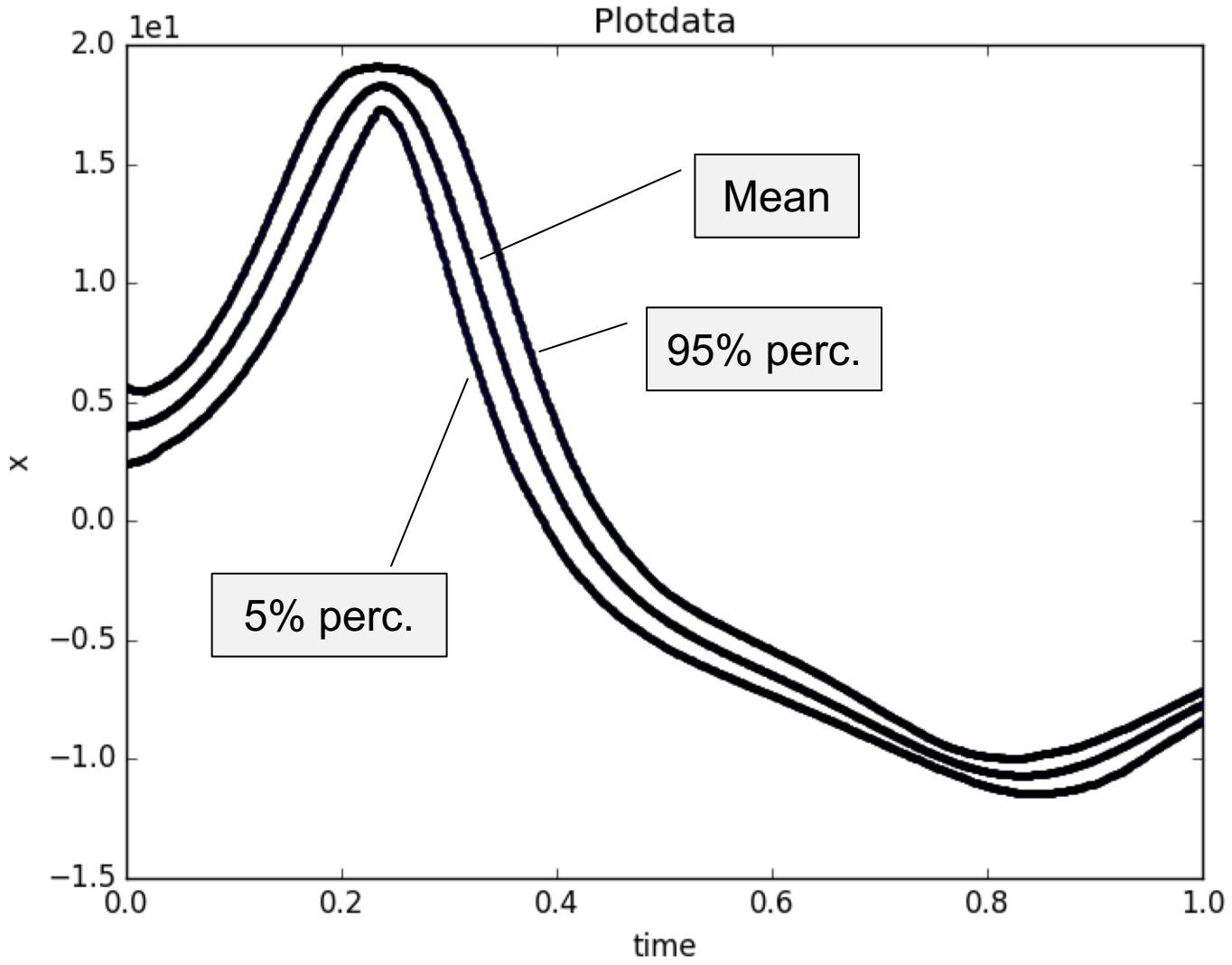
```
<Steps>
  <MultiRun name="FirstMRun">
    <Input    class="DataObjects"   type="PointSet"         >inputPlaceHolder</Input>
    <Model    class="Models"        type="ExternalModel"    >PythonModule</Model>
    <Sampler  class="Samplers"      type="MonteCarlo"       >MC_external</Sampler>
    <Output   class="DataObjects"   type="HistorySet"       >HistorySet</Output>
  </MultiRun>
  <PostProcess name="timeDepBasicStatPP">
    <Input    class="DataObjects"   type="HistorySet"       >HistorySet</Input>
    <Model    class="Models"        type="PostProcessor"    >timeDepBasicStat</Model>
    <Output   class="DataObjects"   type="HistorySet"       >basicStatHistory</Output>
    <Output   class="OutStreams"    type="Plot"             >Plotdata</Output>
    <Output   class="OutStreams"    type="Plot"             >PlotRawdata</Output>
  </PostProcess>
</Steps>
```

Input file name: time_dep.xml

RAVEN Example 1: Time-Dep. Basic Statistics



RAVEN Example 1: Time-Dep. Basic Statistics



RAVEN Example 2
Time Slice Clustering

RAVEN Example 2: Time Slice Clustering

- Steps:
  1. Generate time-dependent data
  2. Cluster time-dependent data

RAVEN Example 2: Time Slice Clustering

Distributions	Models	Samplers	Databases	DataObjects	Steps

```
<Steps>
  <MultiRun name="FirstMRun">
    <Input     class="DataObjects"   type="PointSet"         >inputPlaceHolder</Input>
    <Model     class="Models"        type="ExternalModel"    >PythonModule</Model>
    <Sampler   class="Samplers"      type="MonteCarlo"       >MC_external</Sampler>
    <Output    class="DataObjects"   type="HistorySet"       >outMC</Output>
  </MultiRun>
  <PostProcess name="clustering">
    <Input     class="DataObjects"   type="HistorySet"       >outMC</Input>
    <Model     class="Models"        type="PostProcessor"    >KMeans1</Model>
    <SolutionExport class="DataObjects"  type="HistorySet" >clusterInfo
                   </SolutionExport>
    <Output    class="DataObjects"   type="HistorySet"       >outMC</Output>
  </PostProcess>
</Steps>
```

Input file name: timeSliceKMeans.xml

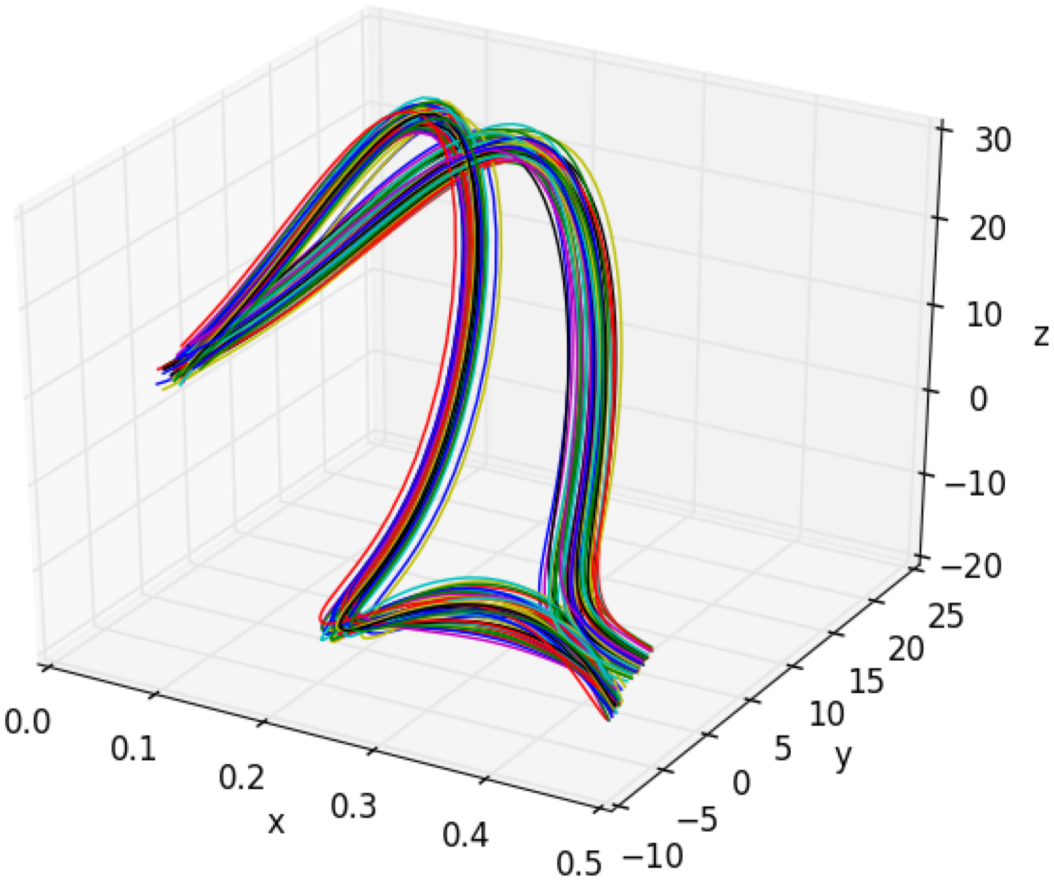
RAVEN Example 2: Time Slice Clustering

Distributions	Models	Samplers	Databases	DataObjects	Steps

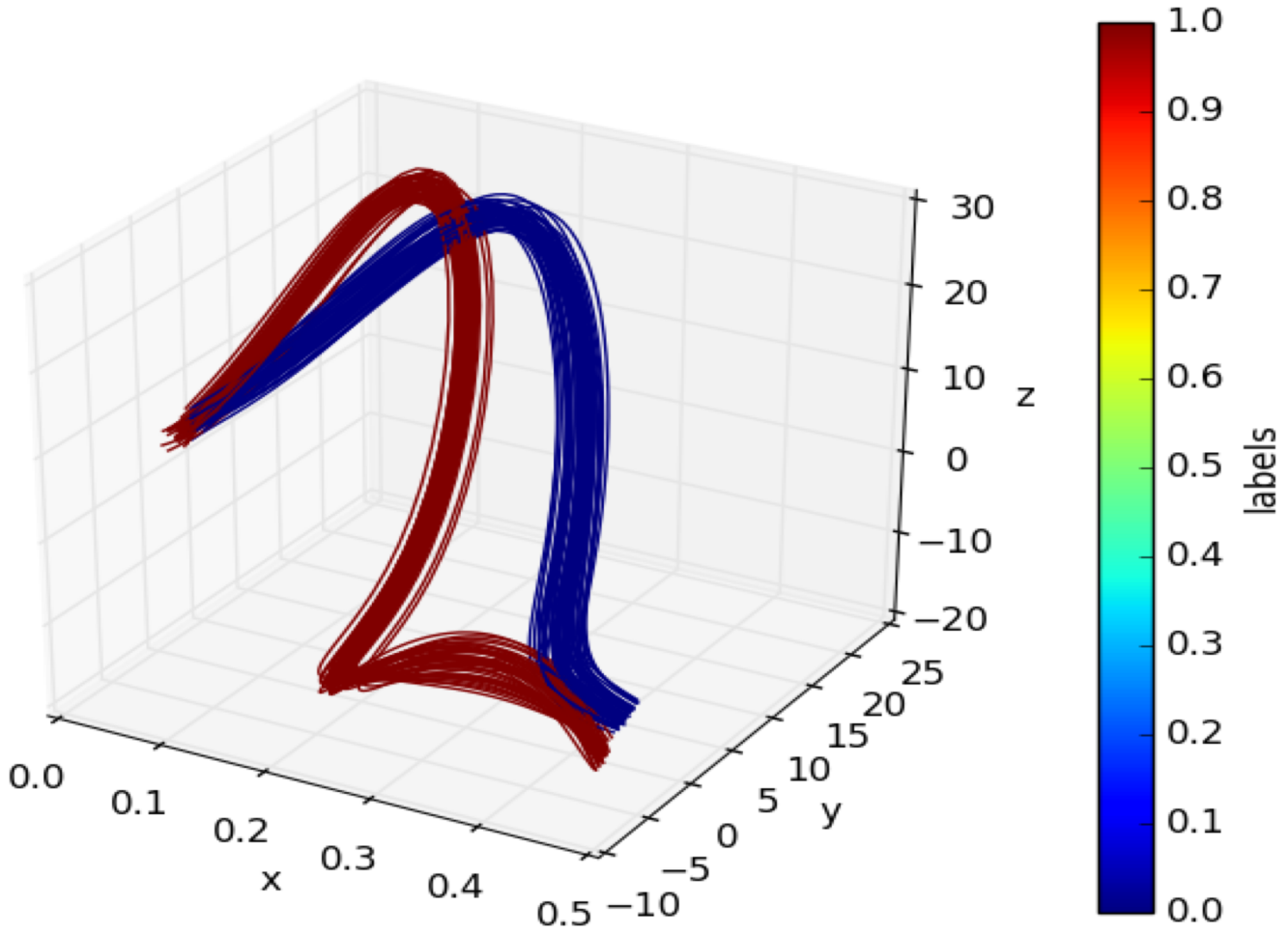
```
<Models>
  ...
  <PostProcessor name="KMeans1" subType="DataMining">
    <KDD lib="SciKitLearn"  labelFeature="labels">
      <SKLtype>cluster|KMeans</SKLtype>
      <Features>x,y,z</Features>
      <n_clusters>2</n_clusters>
      <max_iter>1000</max_iter>
      <random_state>1</random_state>
      <init>k-means++</init>
    </KDD>
    <pivotParameter>time</pivotParameter>
  </PostProcessor>
</Models>
```

Input file name: timeSliceKMeans.xml

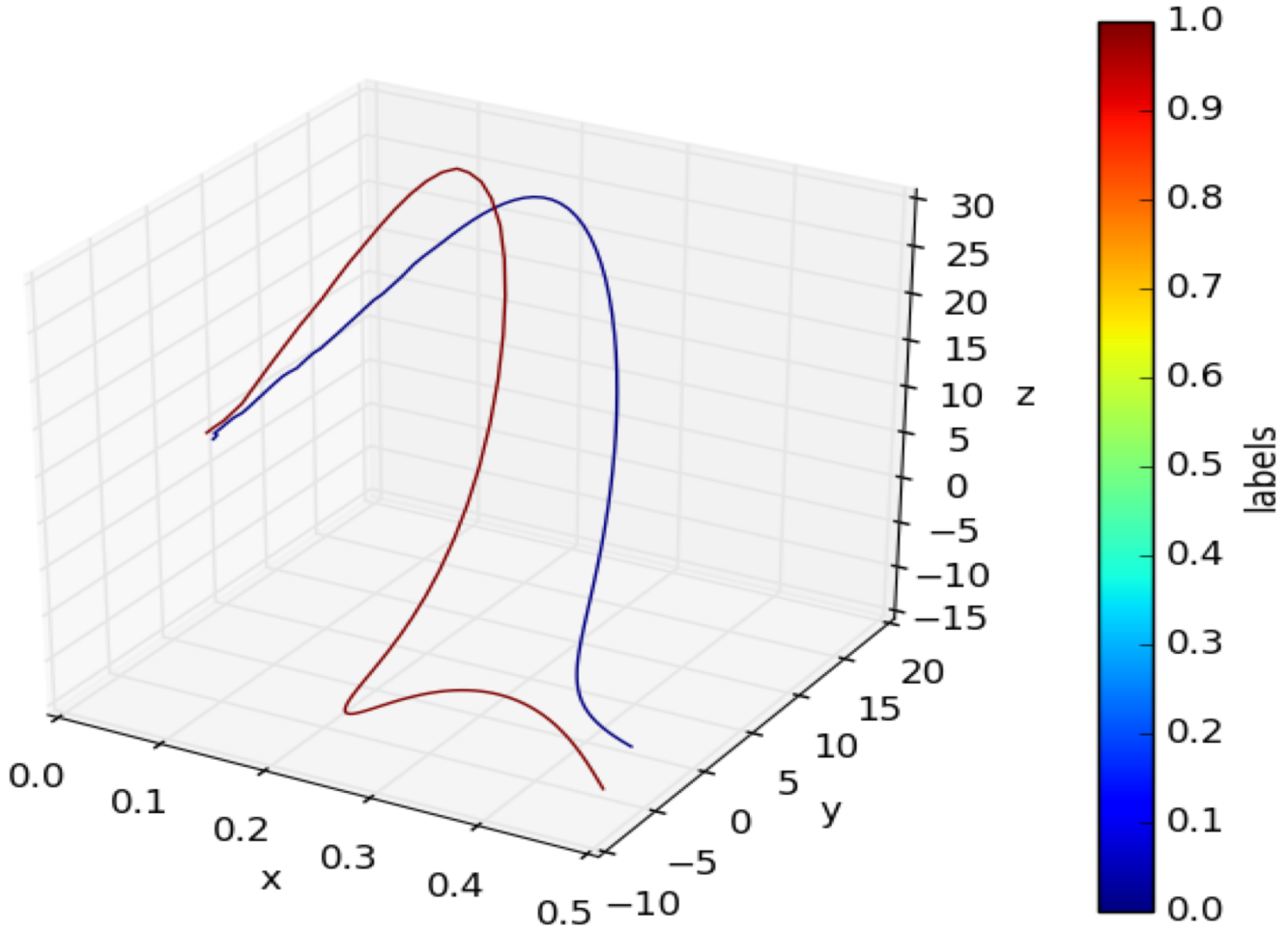
RAVEN Example 2: Time Slice Clustering



RAVEN Example 2: Time Slice Clustering



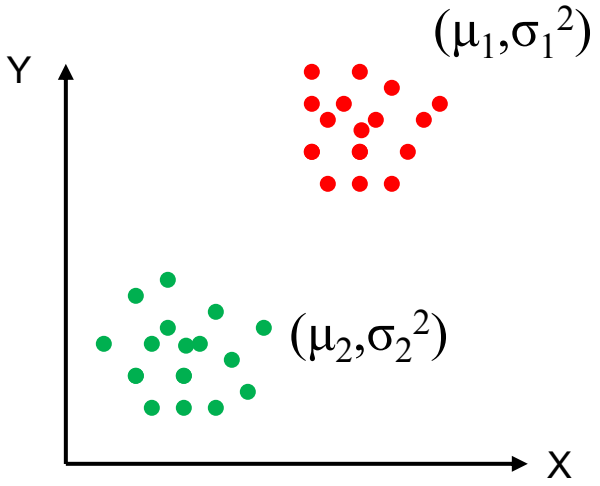
RAVEN Example 2: Time Slice Clustering

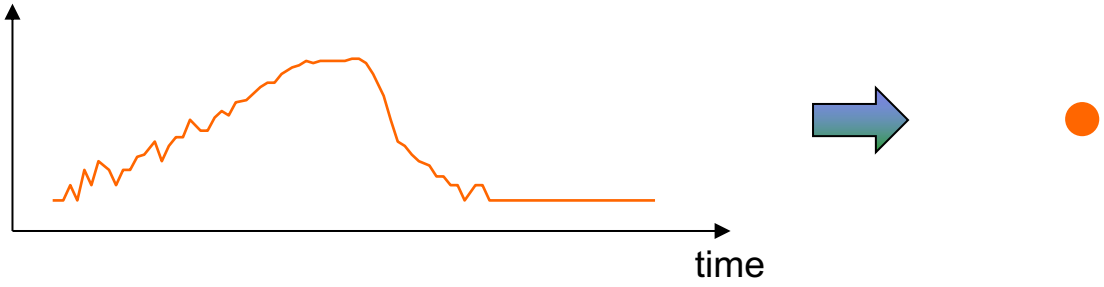


RAVEN Example 3
Time Dependent Clustering
Approach 1: Time Series Transformation

RAVEN Example 3: Time Dependent Clustering (1)

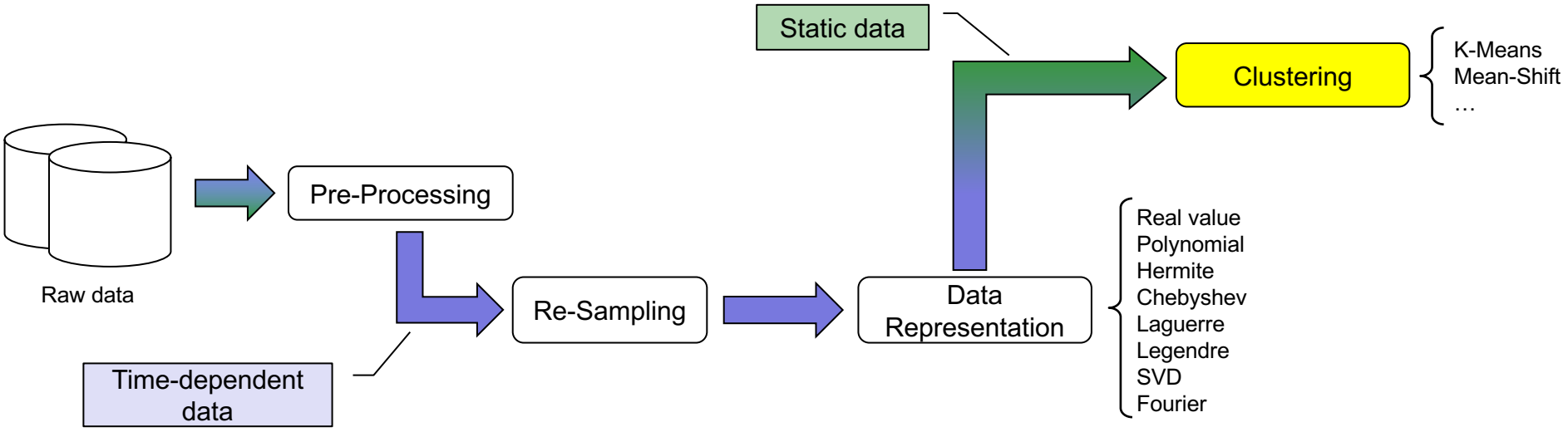
- Objective
  - Analyze time series not at each time step
  - Consider the whole time series as a whole
- Recall clustering
  - Operate on PointSets
- Consider each time series as a “data-point”





RAVEN Example 3: Time Dependent Clustering (1)

- Approach:
  - Convert each time series as a multi-dimensional vector
    - Convert HistorySet into PointSet
  - Perform Clustering
  - Convert clustering results from PointSet to HistorySet



RAVEN Example 3: Time Dependent Clustering (1)

Distributions	Models	Samplers	Databases	DataObjects	Steps

```
<Models>
  ...
  <PostProcessor name="dataPreProc" subType="InterfacedPostProcessor">
    <method>HS2PS</method>
    <pivotParameter>time</pivotParameter>
  </PostProcessor>
  <PostProcessor name="KMeans1" subType="DataMining">
    <PreProcessor class="Models" type="PostProcessor">dataPreProc</PreProcessor>
    <KDD lib="SciKitLearn">
      <SKLtype>cluster|KMeans</SKLtype>
      <Features>output</Features>
      <n_clusters>2</n_clusters>
      <tol>1E-10</tol>
      <random_state>1</random_state>
      <init>k-means++</init>
      <precompute_distances>True</precompute_distances>
    </KDD>
  </PostProcessor>
</Models>
```

Input file name: TD_KMeans.xml

RAVEN Example 3: Time Dependent Clustering (1)

Distributions	Models	Samplers	Databases	DataObjects	Steps

```
<Steps>
  <MultiRun name="FirstMRun">
    <Input     class="DataObjects"   type="PointSet"         >inputPlaceHolder</Input>
    <Model     class="Models"        type="ExternalModel"    >PythonModule</Model>
    <Sampler   class="Samplers"      type="MonteCarlo"       >MC_external</Sampler>
    <Output    class="DataObjects"   type="HistorySet"       >outMC</Output>
  </MultiRun>
  <PostProcess name="clustering">
    <Input     class="DataObjects"   type="HistorySet"       >outMC</Input>
    <Model     class="Models"        type="PostProcessor"    >KMeans1</Model>
    <SolutionExport class="DataObjects"  type="HistorySet" >clusterInfo
                   </SolutionExport>
    <Output    class="DataObjects"   type="HistorySet"       >outMC</Output>
  </PostProcess>
</Steps>
```

Input file name: TD_KMeans.xml

RAVEN Example 3: Time Dependent Clustering (1)

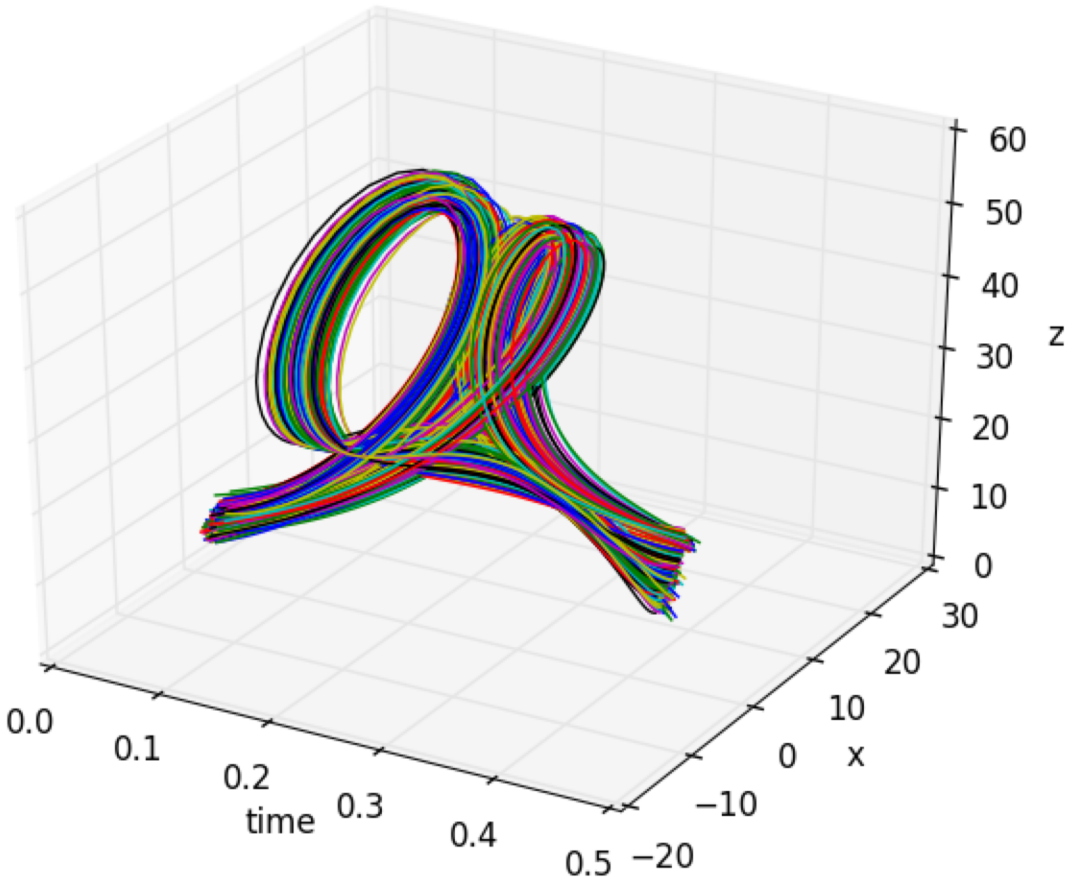
Distributions	Models	Samplers	Databases	DataObjects	Steps

```
<Steps>
  <PostProcess name="filter0">
    <Input    class="DataObjects"   type="HistorySet"      >outMC</Input>
    <Model    class="Models"        type="PostProcessor"   >filter0</Model>
    <Output   class="DataObjects"   type="HistorySet"      >outMC0</Output>
  </PostProcess>
</Steps>
```

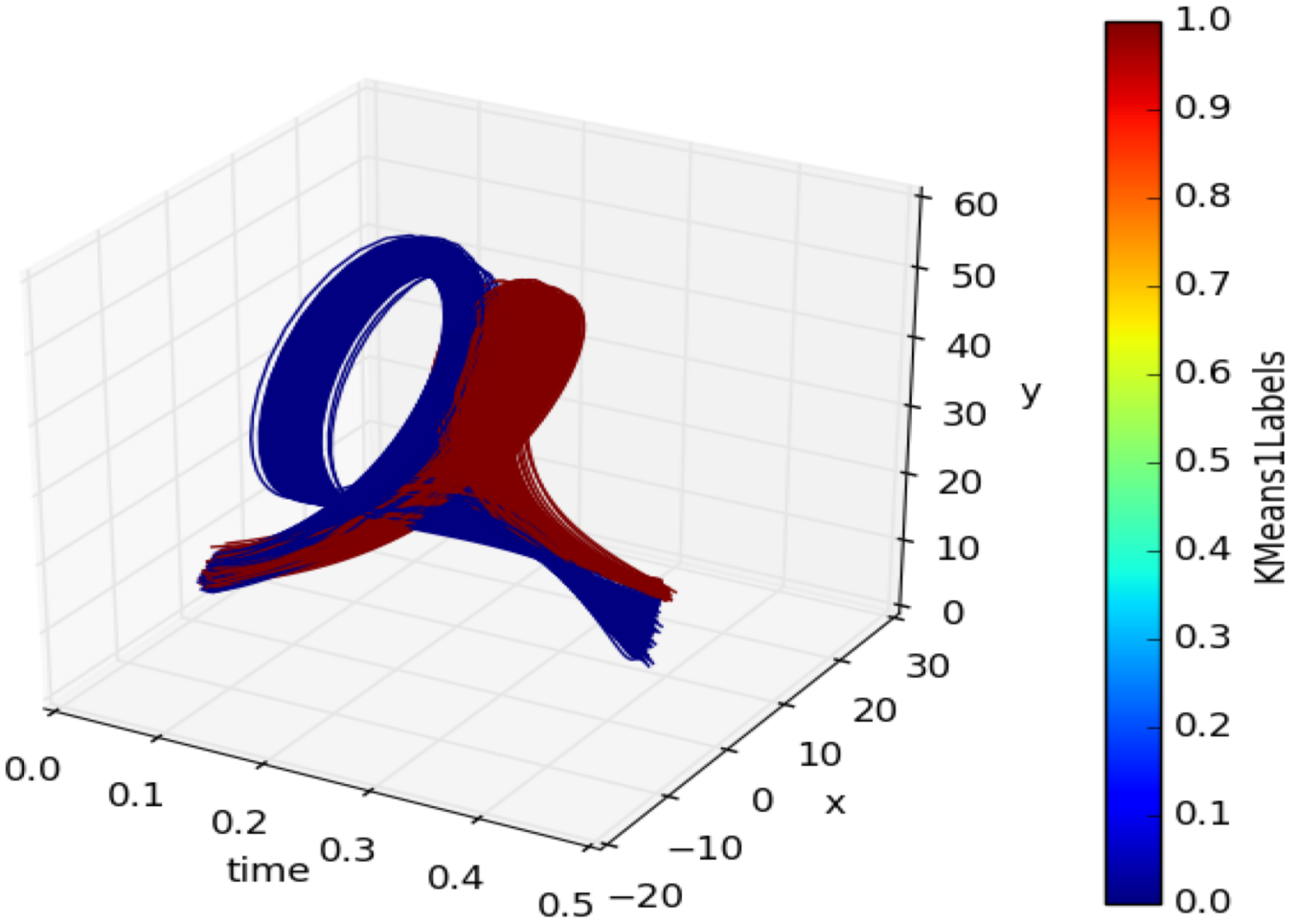
```
<Models>
  ...
  <PostProcessor name="filter0" subType="InterfacedPostProcessor">
    <method>dataObjectLabelFilter</method>
    <dataType>HistorySet</dataType>
    <label>KMeans1Labels</label>
    <clusterIDs>0</clusterIDs>
  </PostProcessor>
</Models>
```

Input file name: TD_KMeans.xml

RAVEN Example 3: Time Dependent Clustering (1)



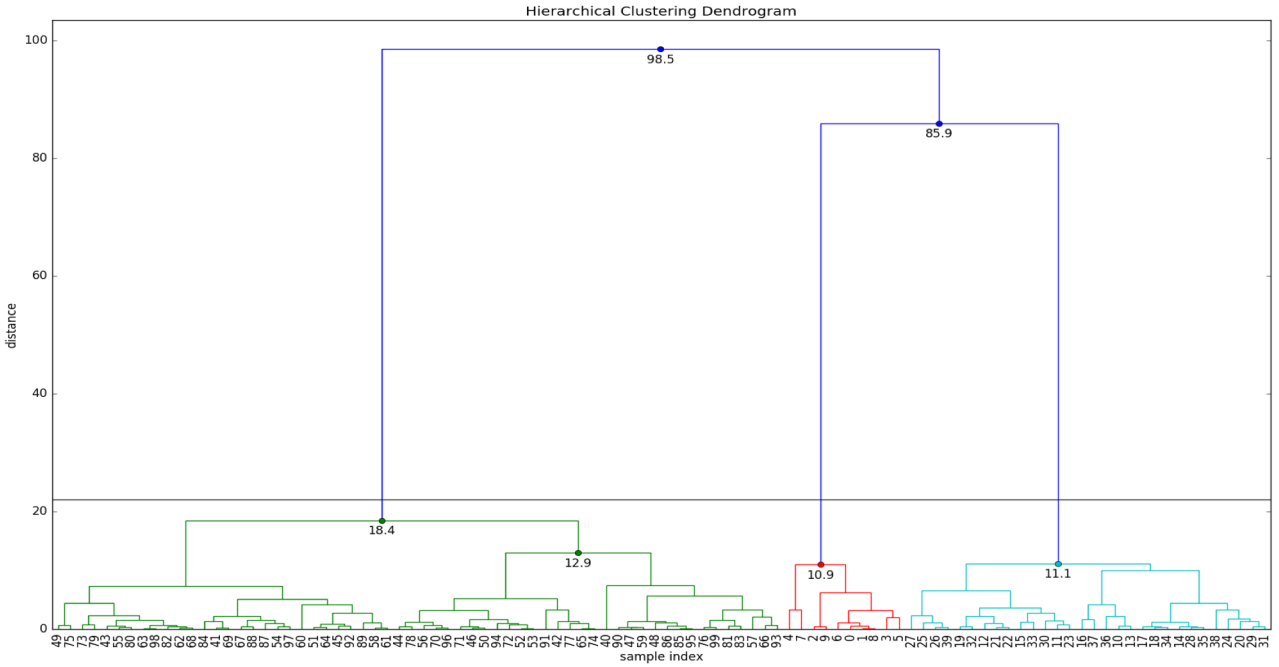
RAVEN Example 3: Time Dependent Clustering (1)

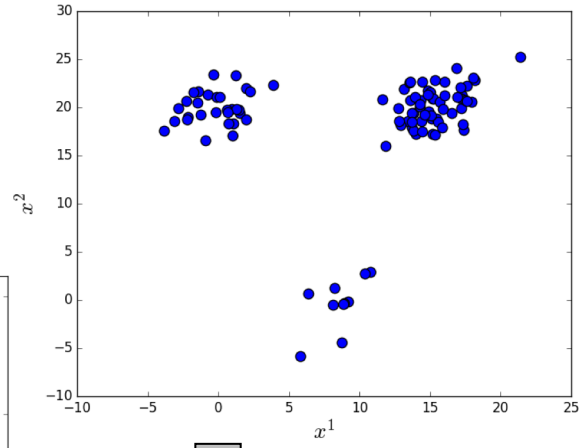


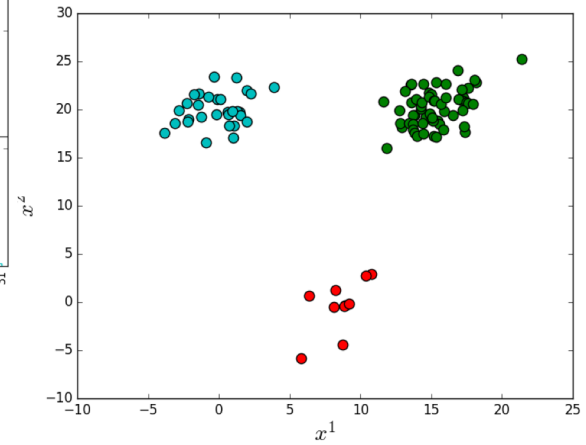
RAVEN Example 4
Time Dependent Clustering
Approach 2: Time Dependent Metrics

RAVEN Example 4: Time Dependent Clustering (2)

- Few clustering algorithms accepts as input a distance metric
  - E.g.: Hierarchical clustering





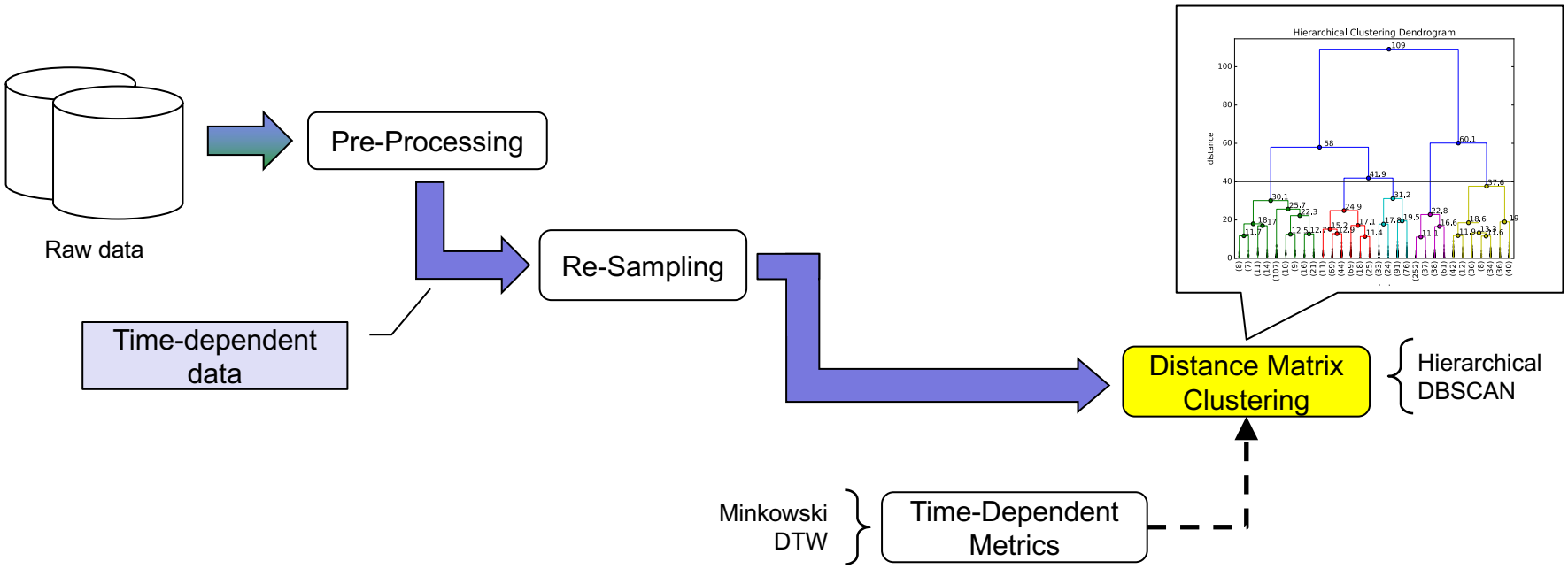


RAVEN Example 4: Time Dependent Clustering (2)

- Distance matrix based clustering
  - Given *N* time series
  - Input is a *N*x*N* dimensional matrix Δ

$$\Delta = [d(S,Q)]$$

$d(S,Q)$ is the distance between the S^{th} and Q^{th} time series



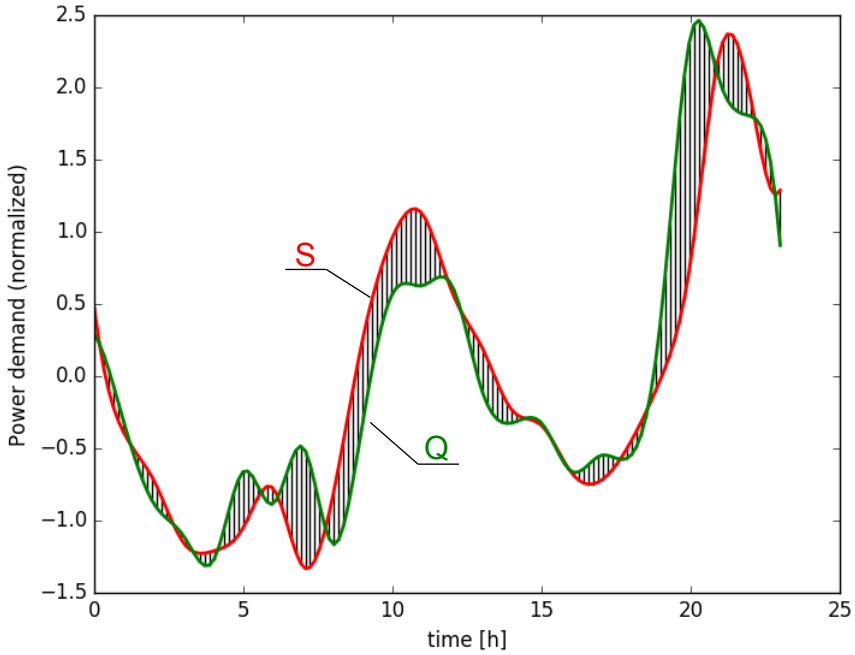
RAVEN Example 4: Time Dependent Clustering (2)

- Euclidean distance

$$S = s_0, s_2, \ldots, s_T$$

$$Q = q_0, q_2, \ldots, q_T$$

$$d^{Eucl.}(S,Q) = \sqrt{\sum_{t=0}^{T}(s_i - q_i)^2}$$



- The good
  - Fast computation

- The bad
  - Sensitive to offset translation (time delays)
  - Requires time series with identical lengths

RAVEN Example 4: Time Dependent Clustering (2)

- Dynamic Time Warping (DTW)

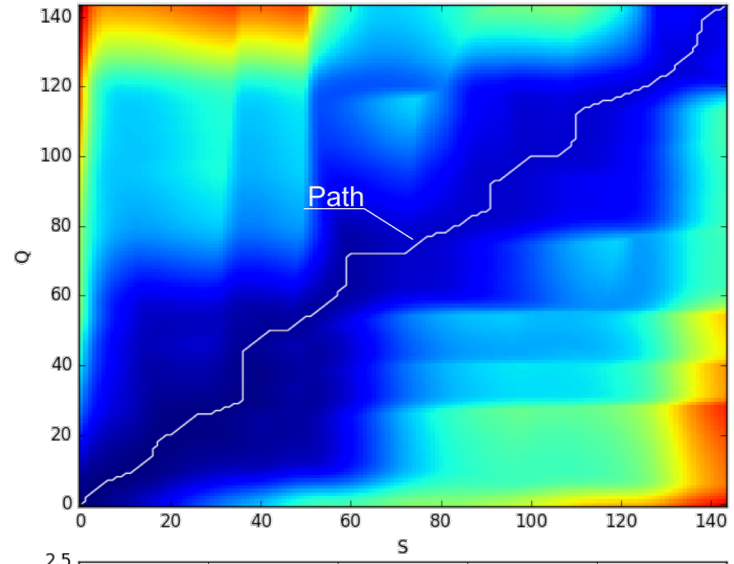
$$S = s_0, s_2, \ldots, s_N$$

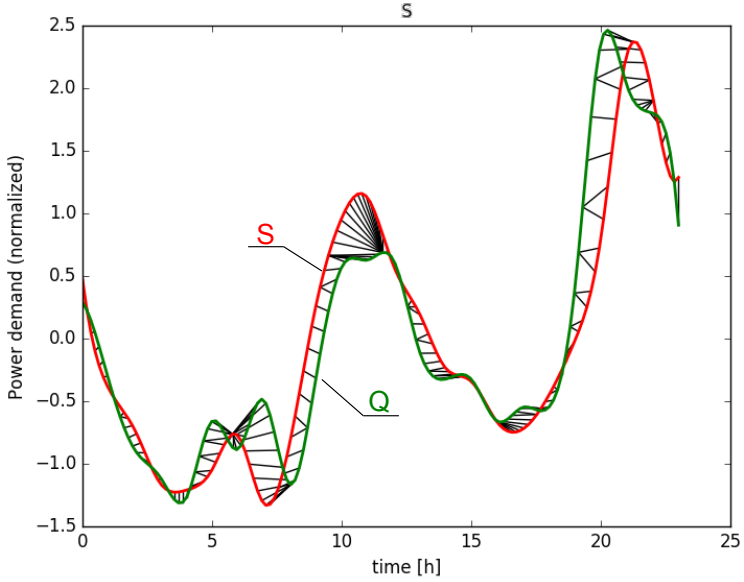
$$Q = q_0, q_2, \ldots, q_M$$

  - Each element of S is linked to the closest element of Q through a global minimization problem

- The good: handling of
  - Small time shifts (i.e., delays)
  - Time series with different time lengths

- The bad: much slower computation





RAVEN Example 4: Time Dependent Clustering (2)

Distributions	Models	Samplers	Metrics	DataObjects	Steps

```
<Metrics>
  <DTW name="example">
    <order>0</order>
    <localDistance>euclidean</localDistance>
    <pivotParameter>time</pivotParameter>
  </DTW>
  <Minkowski name="example">
    <p>2</p>
    <pivotParameter>time</pivotParameter>
  </Minkowski>
</Metrics>
```

Input file name: TD_hierarchical_dtw.xml

RAVEN Example 4: Time Dependent Clustering (2)

Distributions	Models	Samplers	Metrics	DataObjects	Steps

```
<Models>
  ...
  <PostProcessor name="hierarchical" subType="DataMining">
    <Metric class="Metrics" type="DTW">example</Metric>
    <KDD lib="Scipy" labelFeature="labels">
      <Features>output</Features>
      <SCIPYtype>cluster|Hierarchical</SCIPYtype>
      <method>single</method>
      <metric>euclidean</metric>
      <level>2</level>
      <criterion>distance</criterion>
      <dendrogram>true</dendrogram>
      <truncationMode>lastp</truncationMode>
      <p>20</p>
      <leafCounts>True</leafCounts>
      <showContracted>True</showContracted>
      <annotatedAbove>10</annotatedAbove>
    </KDD>
  </PostProcessor>
</Models>
```

Input file name: TD_hierarchical_dtw.xml

RAVEN Example 4: Time Dependent Clustering (2)

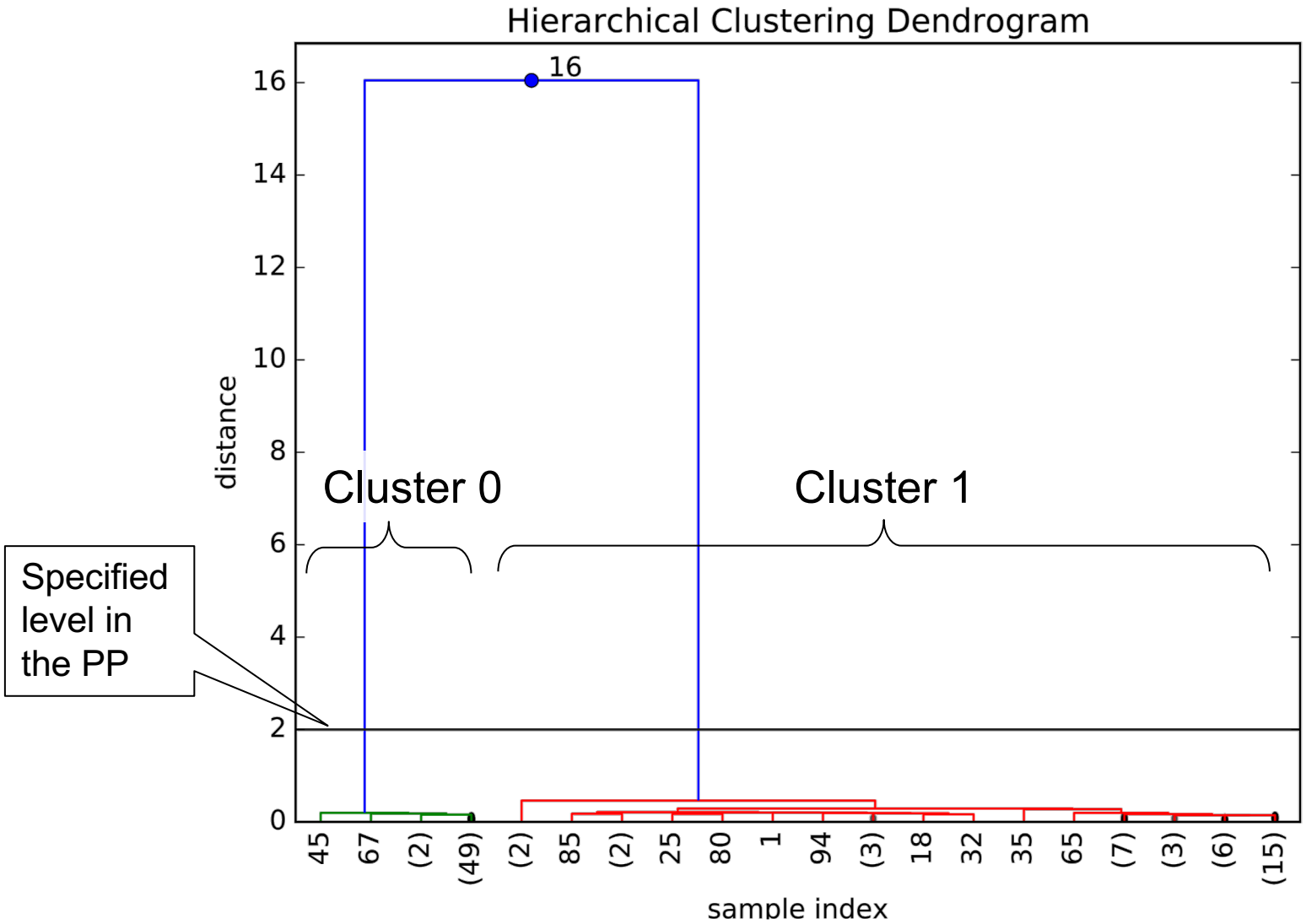
Distributions	Models	Samplers	Metrics	DataObjects	Steps

```
<Steps>
  <MultiRun name="FirstMRun">
    <Input     class="DataObjects"   type="PointSet"         >inputPlaceHolder</Input>
    <Model     class="Models"        type="ExternalModel"    >PythonModule</Model>
    <Sampler   class="Samplers"      type="MonteCarlo"       >MC_external</Sampler>
    <Output    class="DataObjects"   type="HistorySet"       >outMC</Output>
  </MultiRun>
  <PostProcess name="clustering">
    <Input     class="DataObjects"   type="HistorySet"       >outMC</Input>
    <Model     class="Models"        type="PostProcessor"    >hierarchical</Model>
    <Output    class="DataObjects"   type="HistorySet"       >outMC</Output>
  </PostProcess>
</Steps>
```

Note: no solution export

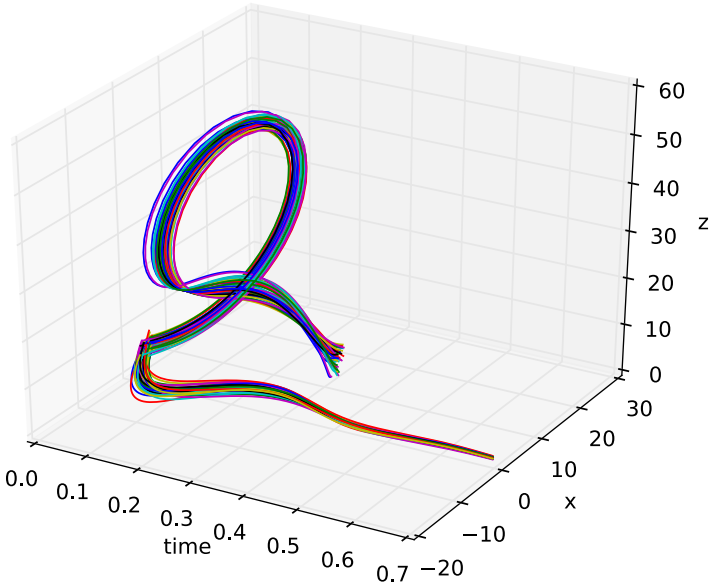
Input file name: TD_hierarchical_dtw.xml

RAVEN Example 4: Time Dependent Clustering (2)



RAVEN Example 4: Time Dependent Clustering (2)

Cluster 0



Cluster 1