Integration and Automation of Data Preparation and Data Mining

Yanhui Geng
Huawei Technologies
Agenda

• Introduction
• Karma – Data Modeling and Integration
• Prediction Task
• Data collection
• Preparing the mode of transportation data
• Using Karma
• Our Approach - Karma Workflow
• Evaluation
• Related Work
• Discussion
Introduction

• Data preparation – To transform the raw data into a form that could be consumed by mining tools

• Raw data collected is heterogeneous, noisy, inconsistent and incomplete

• Data Preparation is an iterative task

• Preparation tasks - cleaning, discretization, transformation and data integration

• Consumes 70 to 80% of the total time
Karma

Interactive tool for rapidly extracting, cleaning, transforming, and publishing data

We propose to combine the steps in data preparation and data mining into a single integrated process using Karma.

Capture detailed metadata about the data sources, transformations and mining services that are invoked.
Predicting the Mode of transportation

- Collect data from GPS and Accelerometer sensors
- Record mode of transport labels
- Extract and transform collected data to generate useful features
- Split the dataset into training and testing sets
- Use Support Vector Machine (SVM) algorithm to train a model with the training data
- Predict mode of transport on records in the testing data
Data Collection

Collected Accelerometer and GPS sensor data using Android App for different modes of transportation
Data collection cont’d

- Total 3 days data was collected
- For each day we have 3 csv files
  - AccelerometerSensor.csv
  - LocationProbe.csv
  - TransportationLabels.csv
- User manually noted the time period for each mode of transportation used
Preparing the mode of transportation data

Extract & transform fields from Accelerometer data

Add DFT energy coefficients for 1Hz, 2Hz & 3Hz

Extract & transform fields from Location(GPS) data

Join GPS data with DFT coefficients

Label the rows using Transportation Labels and Timing information

<table>
<thead>
<tr>
<th>timestamp</th>
<th>speed</th>
<th>accuracy</th>
<th>acceleration magnitude</th>
<th>DFT_E1</th>
<th>DFT_E2</th>
<th>DFT_E3</th>
<th>mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1387869469</td>
<td>0</td>
<td>16</td>
<td>11.69130897</td>
<td>136.686705</td>
<td>139.957767</td>
<td>139.957767</td>
<td>walking</td>
</tr>
<tr>
<td>1388062990</td>
<td>0.89422005</td>
<td>8</td>
<td>11.8207537</td>
<td>139.730218</td>
<td>139.730218</td>
<td>135.891275</td>
<td>stationary</td>
</tr>
<tr>
<td>1388060907</td>
<td>2.3307722</td>
<td>12</td>
<td>12.17176955</td>
<td>148.151974</td>
<td>148.151974</td>
<td>146.537468</td>
<td>bus</td>
</tr>
<tr>
<td>1388059088</td>
<td>7.702458</td>
<td>12</td>
<td>14.09193116</td>
<td>198.582524</td>
<td>92.5838217</td>
<td>104.223227</td>
<td>auto</td>
</tr>
</tbody>
</table>
Using Karma

- These tasks are performed only the first time
- Modeling the raw datasets and the required web services
- All transformations and processing done here is recorded by Karma

- The Karma execution tasks are ones that are repeated for each dataset.
- Applying transformations, join operations and invoking the data mining services
Workflow using Karma

Data Collection

Karma

Step 1: Modeling Data and Services

Step 2: Data Preparation

Step 3: Data Mining
Workflow using Karma

1. Data Collection
2. Step 1: Modeling Data and Services
3. Step 2: Data Preparation
4. Step 3: Data Mining

Karma

Karma Setup

Karma Execution
Data Collection

- Sensor data for Accelerometer and GPS
- Transportation Labels

Karma Step1: Modeling Data and Services

- LocationProbe
- AccelerometerSensor
- DFT Calculation Service
- Labeling Service
- SVM Training Service
- SVM Testing Service
Workflow cont’d
Karma Step 1: Modeling Data and Services

Applying a Semantic Model to the data set

[Diagram showing relationships between Accelerometer Reading, Mode of Transport, Motion Sensor, DFT Coefficients (DFT_E1, DFT_E2, DFT_E3), and other properties like magnitude, speed, accuracy, hasValue, hasMovement, hasCoefficients, and timestamp.]
Modeling the LocationSensor Data

- Round off the timestamp column using Python transform
- We model only the required columns - timestamp, accuracy and speed and add URLs for both the classes using the timestamp values
- Publish the RDF
Modeling the DFT service

- Calculate “Magnitude” using a Python transformation as
  \[ \text{magnitude} = \sqrt{x^2 + y^2 + z^2} \]

- Set semantics for the timestamp and magnitude columns

- Set additional properties like service url, method, etc. and publish the model
Workflow cont’d

Karma Step 1: Modeling Data and Services

Karma Step 2: Data Preparation

- Process Location Probe data
- Load Accelerometer Sensor data
- Pytransform for Acceleration Magnitude
- Extract timestamp and magnitude columns
- Invoke addDFT service
- Join addDFT output and Location Probe data
- Filter rows that cannot be joined
- Add mode of transportation labels

Karma Step 3: Data Mining
Workflow cont’d

Karma Step 2: Data Preparation

Processing Accelerometer files

• Apply the ‘AccelerometerSensor’ model and publish the data

• Invoke the DFT service. The DFT service produces a new worksheet which contains the new columns for DFT coefficients
Workflow cont’d
Karma Step 2: Data Preparation

- Add the url for ‘AccelerometerReading’ class
- Publish the data
- Join the data with the location dataset

- Invoke the labeling service on the augmented dataset
Workflow cont’d

Karma Step 2: Data Preparation

Karma Step 3: Data Mining

Invoke SVM Training service

Invoke SVM Testing service

Train & Update SVM models

SVM Training Summary

SVM Prediction output
Workflow cont’d
Karma Step 3: Data Mining

• Karma automatically identifies which services can be invoked on the current data

• Karma matches the semantic types and the relationship between the classes of the data with all the service models in the repository

• A list of services is shown to the user along with the number of properties it uses as inputs for the service
Workflow cont’d
Karma Step 3: Data Mining

How Karma identifies services that could be invoked on the data set

DFT Service Model

ModeOfTransport

Data Model

Acceleration

Timestamp | Magnitude
---|---

Acceleration

Timestamp | Magnitude | Timestamp
---|---|---

Workflow cont’d
Karma Step 3: Data Mining

How Karma identifies services that could be invoked on the data set

Karma matches the class and semantic types and determines that the DFT service can be invoked
Workflow cont’d
Karma Step 3: Data Mining

Karma interface with data mining services

Data mining algorithms in R
- SVM
- Decision Trees

Java REST service

Model Repository

Karma

JSON, XML, CSV
Karma Step 3: Data Mining

- Karma can interact with a web service using the service model
- In our current example, the SVM is implemented in R programming language
- A Java based REST service is used as an interface for the R programs
- The REST service keeps tracks of all the models that were trained using a unique model identifier
Evaluation

- We evaluated our approach by measuring
  - Reduction in the time and
  - Reduction in effort required to perform data preparation and data mining for the mode of transport prediction task
- We compared the time taken using Karma and MS Excel
- The effort and time to write scripts for DFT calculation, SVM, etc. were excluded as they were part of both approaches
Evaluation cont’d
Using MS Excel

1. Merge the LocationProbe.csv file from each day into a single file
2. Processing AccelerometerSensor.csv
   1. Transform Timestamp column
   2. Calculate Magnitude for each row in a new column
   3. Save in a new file
3. Invoke python script for DFT calculations on the previous file
4. Processing LocationProbe.csv
   1. Extract Timestamp, Accuracy and Speed columns in a new sheet
   2. Transform Timestamp column
   3. Join the output of DFT calculation script with the LocationProbe file to attach Speed and Accuracy columns.
   4. Save the file
5. Invoke the python script for labeling the joined data
6. Invoke the SVM training script
## Evaluation cont’d

Time taken by Karma for one trial of data processing and data mining

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>User Time (sec)</th>
<th>System Processing Time (sec)</th>
<th>Total Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modeling LocationProbe data</td>
<td>34</td>
<td>18</td>
<td>0:52</td>
</tr>
<tr>
<td>2</td>
<td>Publish RDF for LocationProbe</td>
<td>12</td>
<td>6</td>
<td>1:10</td>
</tr>
<tr>
<td>3</td>
<td>Modeling AccelerometerSensor data</td>
<td>18</td>
<td>5</td>
<td>1:34</td>
</tr>
<tr>
<td>4</td>
<td>Publish RDF for AccelerometerSensor</td>
<td>11</td>
<td>9</td>
<td>1:54</td>
</tr>
<tr>
<td>5</td>
<td>Invoke addDFT service</td>
<td>8</td>
<td>2</td>
<td>2:04</td>
</tr>
<tr>
<td>6</td>
<td>Modeling DFT service output</td>
<td>10</td>
<td>2</td>
<td>2:16</td>
</tr>
<tr>
<td>7</td>
<td>Publish RDF for DFT output</td>
<td>11</td>
<td>6</td>
<td>2:33</td>
</tr>
<tr>
<td>8</td>
<td>Join with LocationProbe RDF</td>
<td>12</td>
<td>5</td>
<td>2:50</td>
</tr>
<tr>
<td>9</td>
<td>Publish the augmented model</td>
<td>15</td>
<td>3</td>
<td>3:08</td>
</tr>
<tr>
<td>10</td>
<td>Publish RDF for joined data</td>
<td>10</td>
<td>6</td>
<td>3:24</td>
</tr>
<tr>
<td>11</td>
<td>Invoke getLabel service</td>
<td>8</td>
<td>2</td>
<td>3:34</td>
</tr>
<tr>
<td>12</td>
<td>Filter our ‘NA’ mode of transport</td>
<td>31</td>
<td>3</td>
<td>4:08</td>
</tr>
<tr>
<td>12</td>
<td>Model mode of transport data - the result of add label service</td>
<td>6</td>
<td>3</td>
<td>4:17</td>
</tr>
<tr>
<td>13</td>
<td>Publish RDF for Model of transport data</td>
<td>20</td>
<td>4</td>
<td>4:41</td>
</tr>
</tbody>
</table>
## Evaluation cont’d

### Time taken by MS Excel for one trial of data processing and data mining

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>User Time (sec)</th>
<th>System Processing Time (sec)</th>
<th>Total Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Process AccelerometerSensor data — add magnitude and set timestamp column to be 4 decimal places</td>
<td>44</td>
<td>0</td>
<td>0:44</td>
</tr>
<tr>
<td>2</td>
<td>Extract timestamp and Magnitude in new worksheet and save as CSV</td>
<td>41</td>
<td>0</td>
<td>1:25</td>
</tr>
<tr>
<td>3</td>
<td>Invoke addDFT script</td>
<td>8</td>
<td>2</td>
<td>1:35</td>
</tr>
<tr>
<td>4</td>
<td>Process addDFT output file — format timestamp column to be 4 decimal places</td>
<td>12</td>
<td>0</td>
<td>1:48</td>
</tr>
<tr>
<td>5</td>
<td>Copy timestamp, speed and accuracy columns from LocationProbe data into a new worksheet</td>
<td>41</td>
<td>0</td>
<td>2:29</td>
</tr>
<tr>
<td>6</td>
<td>Process timestamp column to be 4 decimal places, and add a new column to round off the decimal</td>
<td>25</td>
<td>0</td>
<td>2:54</td>
</tr>
<tr>
<td>7</td>
<td>Add vLookUp formulae in the AccelerometerData worksheet for Speed</td>
<td>27</td>
<td>0</td>
<td>3:21</td>
</tr>
<tr>
<td>8</td>
<td>Add vLookUp formulae in the AccelerometerData worksheet for Accuracy</td>
<td>34</td>
<td>0</td>
<td>3:55</td>
</tr>
<tr>
<td>9</td>
<td>Apply filter to remove unmatched — NA rows after join and delete them.</td>
<td>43</td>
<td>0</td>
<td>4:38</td>
</tr>
<tr>
<td>10</td>
<td>Save this accelerometer with DFT data for input to labeling service</td>
<td>19</td>
<td>0</td>
<td>4:57</td>
</tr>
<tr>
<td>11</td>
<td>Invoke the labeling service over the exported CSV file</td>
<td>12</td>
<td>1</td>
<td>5:09</td>
</tr>
<tr>
<td>12</td>
<td>Filter data to remove NA columns</td>
<td>32</td>
<td>0</td>
<td>5:41</td>
</tr>
<tr>
<td>13</td>
<td>Save the file as ProcessedData file</td>
<td>6</td>
<td>0</td>
<td>5:48</td>
</tr>
<tr>
<td>14</td>
<td>Copy the ProcessedData file to the required location for SVM invocation</td>
<td>10</td>
<td>0</td>
<td>5:58</td>
</tr>
</tbody>
</table>
Evaluation cont’d

• We performed two trials of data preparation and data mining
• Each trial consisted of 3 days data
• Accuracy is 100% for both approaches because the user can always go back and rectify an error in data preparation or data mining
### Evaluation cont’d

<table>
<thead>
<tr>
<th></th>
<th>Karma</th>
<th>MS Excel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total time for trail 1 and 2</td>
<td>22:39 min</td>
<td>40:20 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Reduction excluding karma setup</td>
<td>17:41 min</td>
<td>42.14%</td>
</tr>
<tr>
<td>Total Reduction including karma setup (Set up time : 9:30 min)</td>
<td>8:11 min</td>
<td>20.28%</td>
</tr>
<tr>
<td>Accuracy with Karma</td>
<td>100.00%</td>
<td></td>
</tr>
<tr>
<td>Accuracy with Excel</td>
<td>100.00%</td>
<td></td>
</tr>
</tbody>
</table>
Related Work

- **RapidMiner and KNIME**
  - have data preparation features
  - have support for invoking remote web services
  - offer integration of data preparation and mining
  - lack semantic definition of remote web services that can be published and shared

- **DataPreparator and Google Refine**
  - have data preparation features
  - do not offer integration of data preparation and mining

- **Our Work**
  - offers all the above features (bulleted with ‘+’)
  - offers semantic definition of remote web services
  - offers automation of data preparation tasks
Discussion

• An end-to-end approach of data preparation and data mining
• Ability to share models across users by using semantic web technology
• Users need not be an expert in machine learning or have advanced programming skills to perform data mining

http://www.isi.edu/integration/karma
Thank You