Sticks to Structures: Using Logistic Regression to Map an Ambiguous Schema to XML

C. Aaron Rodgers, Daniel Dean, Hui Lei, Jason Gu, Dingcheng Li, Pei Ni Liu, Xiao Xi Liu, YiFan Du
Bio: C. Aaron Rodgers

- Third-year Ph.D. student in the Computer Science Department at Purdue University.
- Member of Dr. Walid Aref's Database Research Group.
- Background in relational databases.
- Current research interests are spatio-temporal indexing, cluster-based data management tools and algorithms (HDFS, MapReduce, etc.).
Outline

- Context.
- Problem.
- Related Work.
- Baya.
- Demo.
- Conclusions.
Members of the healthcare industry are increasingly exchanging data digitally.

- Hospital stays.
- Diagnoses
- Prescriptions.
- Insurance information.
- Billing information.
In the U.S., 95% of healthcare organizations use version 2.7 of the Health Level 7 (HL7) medical data exchange standard.

- Version 2.7 released in 2011.
- Maintained by Health Level International.
- Application layer of the OSI model.
Example HL7 Message
Example HL7 Message

MSH|^&|SOARIAN|E_AMC||20151220101811|ADT^A08|154030598|T|2.4|||||
ZSH|SIERX~SYNDRSJR/V|||
EVN|A08|20151220101811|||
PID||8505059^MR|ICDTENPONE^TWENTY%%%%L||19810519000000|M|W|123 TESTING LANE^ALBANY^NY
PD1||33204%%%%Theresa Viola, MD%%%%MSI||HasLivingWill|||
NK1|1|PTEMP||ELLIS HOSPITAL|||
PV1|E|EDWR^EDWR-M62|Unknown||ED||N|ER|||20151220101700|||
PV2|Unknown|^LAST TEST|||Quick Checked In|||
OBX|1|CWE|8661-1^CHIEF COMPLAINT|VALUE:FIND|PT:PATIENT:NM:REPORTED^LN|||LAST TEST|||
OBX|2|CWE|^MEDHISTORYCONSENT||Y^HL70136
Example HL7 Message

MSH|~\&|SOARIAN:E_AMC|||20151220101811||ADT^A08|154030598|T|2.4|||1|
ZSH|SIERX~SYNRDSJRJ|||
E VN|A08|20151220101811|||
P ID|||8505059^MR||CDTENPONE^TWENTY~~~~~L||19810519000000|M|W|123 TESTING LANE^ALBANY^NY
P D1|||33204^^^^^Theresa Viola, MD^MSI||HasLivingWill|||
NK1.1|||PT E MP|||ELLIS HOSPITAL|||
PV1|E|EDWR^EDWR-M62|Unknown|||ED|||N|ER|||
|20151220101700|||
PV2|Unknown|^LAST TEST|||
|Quick Checked In|||
OBX|1|CWE|8661-1^CHIEF COMPLAINT|VALUE:FIND:PT: PATIENT: NOM: REPORTED^LN|^LAST TEST|||
OBX|2|CWE|^MEDHISTORY:CONSENT||Y^HL70136
Example HL7 Message

- **MSH**
  - Segment type: Segment Type
  - Message type: Message Type
  - Field: Field
  - Subfield: Subfield

- **ZSH**
  - Segment type: Segment Type
  - Field: Field

- **EVI**
  - Segment type: Segment Type
  - Field: Field

- **PID**
  - Segment type: Segment Type
  - Field: Field
  - Subfield: Subfield

- **PV1**
  - Segment type: Segment Type
  - Field: Field

- **PV2**
  - Segment type: Segment Type
  - Field: Field

- **OBX1**
  - Segment type: Segment Type
  - Field: Field

- **OBX2**
  - Segment type: Segment Type
  - Field: Field
Outline

- Context.
- **Problem.**
- Related Work.
- Baya.
- Demo.
- Conclusions.
Problem

Healthcare solution providers wish to provide their clients in the medical industry with tools to pool their medical data and mine it for insights. However...

- HL7 is complex.
  - HTTP 1.1 spec is ~120 pages, HL7 spec is ~2,300 pages.
  - Hospital A may populate HL7 fields differently than Hospital B.
  - Hospital A may itself populate HL7 fields in inconsistent manner.
  - No enforcement of the standard.
  - Data could be corrupted or faulty.
Problem

Hospital A HL7 Messages

- Patient
  - Eric
  - Smith
  - Mr.
- Visit
  - Fever
  - Ibuprofen
Problem

Hospital A

- Patient
  - Eric
  - Smith
  - Mr.
  - Visit
  - Fever
  - Ibuprofen

Hospital B

- Patient
  - Suzie
  - Jones
  - Visit
  - Headache

Hospital A HL7 Messages

Hospital B HL7 Messages
Problem

Hospital A HL7 Messages
- Patient
  - Eric
  - Smith
  - Mr.
- Visit
  - Fever
  - Ibuprofen

Hospital B HL7 Messages
- Patient
  - Suzie
  - Jones
- Visit
  - Headache

Analysis on these datasets is difficult and time consuming.
Partial Solution: Map to Unambiguous Schema

Hospital A → HL7 Messages → Unambiguous Schema

Hospital B → HL7 Messages → Unambiguous Schema

Analysis 1...
Analysis 2...
Analysis 3...
Partial Solution: Map to Unambiguous Schema

Common SIF:

- Much simpler (~1000 lines of XML).
- Unambiguous.

```xml
<xsd:complexType name="humanName">
  <xsd:sequence>
    <xsd:element name="given" type="xsd:string" minOccurs="0"/>
    <xsd:element name="middle" type="xsd:string" minOccurs="0"/>
    <xsd:element name="nickName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="prefix" type="xsd:string" minOccurs="0"/>
    <xsd:element name="suffix" type="xsd:string" minOccurs="0"/>
    <xsd:element name="surname" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```
Partial Solution: Map to Unambiguous Schema

Current pipeline looks like this:

HL7 Raw → HL7 XML → Common SIF → Add Pt. Identifiers → UDMH → Analysis
Partial Solution: Map to Unambiguous Schema

Current pipeline looks like this:

HL7 Raw → HL7 XML → Common SIF → Add Pt. Identifiers → UDMH → Analysis

This mapping is costly and brittle.

We currently use a fixed mapping to convert Raw HL7 messages to Common SIF. This is problematic:

- A fixed mapping will break if two hospitals populate HL7 differently.
- Even for a single hospital, HL7 fields may not be populated consistently.
Problem

- Can we use machine learning to improve this process? We would like to:
  - Learn a mapping from HL7 to Common SIF.
  - Recognize if two hospitals are populating HL7 messages differently.
  - Incrementally improve the mapping so that it works for multiple HL7 sources.
Outline

● Context.
● Problem.
● Related Work.
● Baya.
● Demo.
● Conclusions.
Related Work:


- Similar to our approach: Uses logistic regression and user feedback to improve results.
- Different to our approach: Aligns a pair of ontologies (is-a relationships). In our problem, we must align schema to schema.
Related Work:


- Similar to our work: Derives knowledge about a given entity from the enclosing schema (and other sources as well).
- Different to our work: Designed to allow a user to query large amounts of heterogeneous data sources using keyword searches and semantic data.
Outline

● Context.
● Problem.
● Related Work.
● Baya.
● Demo.
● Conclusions.
Proposed Solution: Baya

- Baya is a Java-based command line tool which uses logistic regression to learn a flexible, intelligent mapping from HL7 to Common SIF.
- Rather than create a fixed mapping, we provide Baya with:
  - "Metrics" which Baya can use to analyze HL7 content. (Features)
  - Training. (Can be manual or automated.)
- Once trained, Baya should provide our three desired features:
  - Provide a mapping from HL7 to Common SIF.
  - Recognize if two hospitals are populating HL7 messages differently.
  - Incrementally improve the mapping so that it works for multiple HL7 sources.
Baya: Usage

- Load a target schema.
- Load a message.
- Map tokens from message to schema.
- Repeat message loading and mapping as necessary. (We have a tool that helps automate this.)
Baya: Intuition

- Given a token from an HL7 message, how well does it "fit" a specific field in Common SIF?

```
PV1|1|E|^G^G^^^^G|C|||X^Manning^Steven|
```

Begins with capital letter? 
Contains numbers? 
Alphanumeric? 
Fits a phone number regex? 
Falls in HL7 lastName field? (not determinative)
Baya: Intuition

- For each token, compute a feature vector.
- Combine feature vectors to get a design matrix $X$.

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>9</th>
<th>6</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>13</td>
<td>1</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
<td>12</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>20</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>14</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

- token: "Manning"
- token: "1990-03-05"
Baya: Intuition

- For each token, compute a feature vector.
- Combine feature vectors to get a design matrix $X$. 

![Diagram showing a 5x3 matrix with tokens and target vectors]
Baya: Intuition

- For each token, compute a feature vector.
- Combine feature vectors to get a design matrix $X$.

Question: How do we determine which features are good predictors for a given field?

Answer: Logistic Regression!

Question: How do we determine which features are good predictors for a given field?

Answer: Logistic Regression!
Baya: Intuition

We have identified four categories of features which we can use to "measure" a token:

1. **Token**
   a. How many characters in the token?
   b. Token fits a certain regex? (Phone number, SSN, etc.)
   c. Average value of the field? (BMI, blood pressure, etc.)

2. **Token-Story**
   a. Which HL7 field does the token appear in?
   b. Which line of the story does the token appear on?
   c. How far is the token from the beginning of the line?
   d. Is the first token of the line "OBX"?

3. **Token-SchemaField**
   a. Does the token frequently appear in the schema field? (Good for prescriptions.)
   b. Do syllables in the token frequently appear in the schema field?

4. **Token-SchemaStructure**
   a. Is there a good candidate for token root?
   b. How far is the token from siblings in the schema structure?
Linear Regression: Theory

- Given an independent variable $x$ and a dependent variable $y$, find the line that minimizes the sum of the squares of the vertical distances between each $y_i$ and the line.

**Cost function:**

$$J_{X,y}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

**Iteration:**

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J_{X,y}(\theta)$$

Linear Regression: Theory

After learning an optimal vector $\theta$, we predict new dependent variables like so:

predicted $y$ for $x_{\text{new}} = \theta^T x_{\text{new}} = \begin{pmatrix} .3 & .2 & -.2 & 1.4 \end{pmatrix} \begin{pmatrix} 1 \\ 20 \\ 10 \\ 200 \end{pmatrix} = 282$

parameters ("weights")

predicted_rent

intercept

house_age

num_rooms

sq_ft
Logistic Regression: Theory

- Given an independent variable $x$ and a dependent predictor variable $y$, find the sigmoid which minimizes the sum of the logs of all incorrect probability predictions.

Cost function:

\[
J_{X,y}(\theta) = \frac{-1}{m} \sum_{i=1}^{m} [y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))]
\]

where:

\[
h_\theta = \frac{1}{1 + e^{-\theta^T x_i}}
\]

Gradient descent iteration:

\[
\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)x_{ij}
\]

Image credit: http://www.biostathandbook.com/simplelogistic.html
Logistic Regression: Theory

After learning an optimal vector $\theta$, we predict new dependent variables like so:

$$\text{predicted } y \text{ for } x_{\text{new}} = \frac{1}{1 + e^{-\theta^T x_{\text{new}}}} = \text{logistic}(\begin{bmatrix} -3.1 & .8 & -.4 & 2.2 \\ 1 & .8 & .2 & 3.0 \end{bmatrix}) = 0.98$$

- probability of rain
- intercept
- cloud_cover
- humidity
- days_since_rain

parameters ("weights")
Logistic Regression: Theory

After learning an optimal vector $\theta$, we predict new dependent variables like so:

$$\text{predicted } y \text{ for } x_{\text{new}} = \frac{1}{1 + e^{-\theta^T x_{\text{new}}}} = \text{logistic}([-3.1, .8, -.4, 2.2]) = 0.98$$

Application: Once we have computed a parameter vector, we can iterate over the tokens in an HL7 message and find the token which has the highest probability of belonging in a given field.
Outline

- Context.
- Problem.
- Related Work.
- Baya.
- Demo.
- Conclusions.
Baya: Demo

- Load a schema.
Baya: Demo

- Load a message from Hospital A.
Baya: Demo

- Parse message into tokens.
Baya: Demo

- Map a token from Message to Schema.
- Repeat load message, map.
Baya: Demo

- Build design matrices, train.
Baya: Demo

- After training on Hospital A, predict on a new message from Hospital A.

--- Schema Elements ---
( 0) Person
  (1) personID [31240040] [100.00%]
  (2) humanName
    (3) given [Walter] [100.00%]
    (4) middle [-----] [---.--%]
    (5) nickName [-----] [---.--%]
    (6) prefix [-----] [---.--%]
    (7) suffix [-----] [---.--%]
    (8) surname [Livingstone] [100.00%]
  (9) sex [A] [83.86%]
  (10) dateOfBirth [-----] [---.--%]
  (11) city [Olney] [100.00%]
  (12) state [MD] [100.00%]
  (13) street [abc] [61.50%]
  (14) zipcode [20899] [100.00%]
Baya: Demo

- We've trained on A.
- Evaluate performance on multiple messages from A.
Baya: Demo

- We've trained on A.
- Predict on a message from Hospital B.

<table>
<thead>
<tr>
<th>ID</th>
<th>Field</th>
<th>Value</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>personID</td>
<td>20151220101300</td>
<td>0.13%</td>
</tr>
<tr>
<td>2</td>
<td>humanName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>given</td>
<td>JAY</td>
<td>100.00%</td>
</tr>
<tr>
<td>4</td>
<td>middle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>nickName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>prefix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>suffix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>surname</td>
<td>SUPERJAY</td>
<td>100.00%</td>
</tr>
<tr>
<td>9</td>
<td>sex</td>
<td>M</td>
<td>82.78%</td>
</tr>
<tr>
<td>10</td>
<td>dateOfBirth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>city</td>
<td>ALBANY</td>
<td>100.00%</td>
</tr>
<tr>
<td>12</td>
<td>state</td>
<td>NY</td>
<td>100.00%</td>
</tr>
<tr>
<td>13</td>
<td>street</td>
<td>123 TESTING LANE</td>
<td>78.77%</td>
</tr>
<tr>
<td>14</td>
<td>zipcode</td>
<td>12209</td>
<td>100.00%</td>
</tr>
</tbody>
</table>
Baya: Demo

- We've trained on A.
- Evaluate performance on multiple messages from B.
Baya: Demo

- Now, train over A and B.
- Predict on a message from A.
Baya: Demo

- We've trained over A and B.
- Evaluate performance on multiple messages from A.
Baya: Demo

- We've trained over A and B.
- Predict on a message from B.
Baya: Demo

- We've trained over A and B.
- Evaluate performance on multiple messages from B.
Outline

- Context.
- Problem.
- Related Work.
- Baya.
- Demo.
- Conclusions.
Future Work

- Flesh out bulk loading tool. Current implementation is brittle, doesn't handle repeated fields (or repeated tokens).
- Parsing the HL7 schema is not trivial and may be prohibitively complex.
- Incorporate a text parser like CTakes to handle plaintext data. (e.g., doctor's notes)
- Perform thorough evaluation (precision, accuracy, F1 score).
Conclusions

- We have shown that Baya is capable of:
  - Learning a mapping from HL7 to Common SIF.
  - Recognizing if two hospitals are populating HL7 messages differently.
  - Incrementally improve the mapping so that it works for multiple HL7 sources.

- Contribution: We have identified a set of metrics which are useful for intelligently mapping an ambiguous, incorrectly populated schema to an unambiguous schema.

Thank you!
Questions?