The Protégé-Owl SWRLTab and Temporal Data Mining in Surgery

G Tusch, M O’Connor, T Redmond, R Shankar and A Das

Stanford Medical Informatics

Medical and Bioinformatics Program
School of Computing and Information Systems
Grand Valley State University
Allendale MI

protégé

Outline

• Introduction (An Example of Transplantation Surgery)
• The SPOT Design
• Statistical Aspects
• SPOT in Surgery
• Conclusion

http://www.ladybird.co.uk/favouriteCharacters/spot.html
Suspected Liver Disease

Abnormal liver tests

Acute < 6 months

Hepatic: \hat{\uparrow} \uparrow \text{ALT}
Mixed: \hat{\uparrow} \text{ALT}, \hat{\uparrow} \text{AlkP}

Diagnostic evaluation
1. IgM Anti-HAV
2. HBsAg
3. IgM Anti-HBc
4. Anti-HCV
5. ANA, SMA
6. Monospot, heterophile
7. Ceruloplasmin
8. Alcohol history
9. Drug history

Liver biopsy in acute liver disease:
Reserved for patients in whom the diagnosis remains unclear despite medical evaluation

Cholestatic: \hat{\uparrow} \uparrow \text{AlkP}, \hat{\uparrow} \uparrow \text{GGT}, \hat{\uparrow} \text{ALT}

Diagnostic evaluation
1. AMA
2. Drug history
3. Ultrasound/MRI
4. MRCP/ERCP

Liver biopsy in chronic liver disease:
Often valuable for diagnosis as well as staging and grading liver disease

Chronic > 6 months

Hepatic: \hat{\uparrow} \uparrow \text{ALT}
Mixed: \hat{\uparrow} \text{ALT}, \hat{\uparrow} \text{AlkP}

Diagnostic evaluation
1. HBsAg
2. Anti-HCV
3. Fe saturation, ferritin
4. Ceruloplasmin
5. \alpha,AT
6. ANA, SMA
7. Ultrasound
8. Alcohol history

Cholestatic: \hat{\uparrow} \uparrow \text{AlkP}, \hat{\uparrow} \uparrow \text{GGT}, \hat{\uparrow} \text{ALT}

Diagnostic evaluation
1. Drug history
2. AMA
3. P-ANCA
4. Ultrasound
5. MRCP/ERCP
SPOT and Temporal Abstraction

- **Purpose of SPOT (S - Protégé – OWL/SWRL – Temporal Abstraction):**
  - Mining large clinical databases including exploration of temporal data
  - Example liver transplantation: researcher looks for patients with an unusual pattern of potential complications of the transplanted organ

- **TA is defined as the creation of high-level summaries of time-oriented data**

- **TA is necessary because**
  - clinical databases usually store raw, time-stamped data
  - clinical decisions often require information in high-level terms
The Temporal-Abstraction Task (Shahar)

- **Input:** time-stamped clinical data and relevant events (interventions)
- **Output:** interval-based abstractions
- Identifies past and present trends and states

**Output types:**
- State abstractions (LOW, HIGH)
- Gradient abstractions (INCREASE, DECREASE)
- Rate Abstractions (SLOW, FAST)
- Pattern Abstractions (CRESCEndo)
  - Linear patterns
  - Periodic patterns
Examples of patient courses in liver Tx
Concept: GOT (=AST) increase

![Chart showing GOT increase in patient courses](chart.png)
Tasks and Software

- Estimation of intervals from learning sample: S (R/S-Plus)

- Build high level concepts (Temporal Abstraction):
  Protégé/OWL/SWRL

- Validate intervals: S (R/S-Plus)

- Run abstractions on original database: RASTA?
Learning Concepts from a Subset (Train & Test Data Set)

**INPUT:** Raw Data  
**OUTPUT:** Atomic Intervals (AI)  
**TASK:** Calculate Scores

**INPUT:** Data, AI  
**OUTPUT:** Concept Intervals  
**TASK:** Combine AI

**XenoBase**  
**Oracle**  
**Access**  
**MySQL**

**R**  
**S-Plus**  
**Java**  
**Protégé OWL/SWRL**

**CORE:** R macros  
**USER:** add-ons in R

**CORE:** Concepts = language  
**USER:** Create new concepts

**Searching for Learned Concepts in Database**

**TASK:** - Search for patients with episodes and additional parameters (e.g., survival)
SPOT Structure

SPOT: S - Protégé – OWL/SWRL – Temporal Abstraction

- Read Data from Database
- Generate Intervals / Data Cleansing
- Transform to Valid Time Model
- Java Interface -> Protégé/OWL
- SWRL Building Blocks
- User Creates New Concepts
- Java Interface -> S
- Statistical Evaluations

S (R/S-Plus) -> OWL/SWRL -> S (R/S-Plus)
SPOT Structure (S)

S Part

- Read Data from Database
- Moving Averages and Levels
- Determine Thresholds (Tree)
- Cross Validation
- Remove Gaps <= 2 days
- Transform to Intervals (VTM)
- Java Interface -> Protégé/OWL
- Statistical Evaluations

Interface
Input Data

• Time stamped data in database or time course graph e.g. in Xenobase

• Researcher (user) marks intervals per parameter (e.g. GOT)
  – Several different non-overlapping intervals are allowed, but only one parameter (independence assumption), i.e. mark as “increasing”, “decreasing”, “high”, etc.
  – Interval value is attached to time-stamped parameter value
  – Generate learning and test samples
Data Structure: Clinical Data Example

Clinical tests (variables)

Test labels

Clinical data matrix

Test values

Patients (cases)

Patient IDs

Clinical tests (variables)
An Example Matrix

(not real patient data)

<table>
<thead>
<tr>
<th>dtrans1</th>
<th>trans1</th>
<th>dbili1</th>
<th>bili1</th>
<th>dtrans2</th>
<th>trans2</th>
<th>dbili2</th>
<th>bili2</th>
<th>dtrans3</th>
<th>trans3</th>
<th>dbili3</th>
<th>bili3</th>
<th>group</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.92</td>
<td>0.99</td>
<td>-0.66</td>
<td>-0.66</td>
<td>-0.37</td>
<td>0.18</td>
<td>-1.12</td>
<td>-0.99</td>
<td>-0.39</td>
<td>-0.05</td>
<td>-0.85</td>
<td>-1.02</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.28</td>
<td>-0.01</td>
<td>-0.07</td>
<td>-0.52</td>
<td>-0.34</td>
<td>-0.08</td>
<td>-0.73</td>
<td>-0.84</td>
<td>-0.42</td>
<td>3.00</td>
<td>-0.45</td>
<td>-0.72</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.51</td>
<td>0.75</td>
<td>0.23</td>
<td>2.77</td>
<td>-0.13</td>
<td>-0.11</td>
<td>1.51</td>
<td>2.88</td>
<td>-0.22</td>
<td>-0.21</td>
<td>0.39</td>
<td>2.55</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.66</td>
<td>0.15</td>
<td>-0.59</td>
<td>-1.27</td>
<td>-0.33</td>
<td>0.08</td>
<td>-0.61</td>
<td>-1.30</td>
<td>-0.41</td>
<td>-0.08</td>
<td>-0.35</td>
<td>-1.09</td>
<td>0.00</td>
</tr>
<tr>
<td>0.40</td>
<td>-0.18</td>
<td>-0.56</td>
<td>-1.53</td>
<td>-0.32</td>
<td>-0.14</td>
<td>-0.61</td>
<td>-1.59</td>
<td>-0.40</td>
<td>-0.16</td>
<td>-0.45</td>
<td>-1.49</td>
<td>0.00</td>
</tr>
<tr>
<td>0.92</td>
<td>1.17</td>
<td>-0.33</td>
<td>-1.48</td>
<td>-0.33</td>
<td>0.11</td>
<td>-0.38</td>
<td>-1.46</td>
<td>-0.40</td>
<td>-0.09</td>
<td>-0.28</td>
<td>-1.32</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.28</td>
<td>-0.35</td>
<td>-0.13</td>
<td>1.03</td>
<td>-0.02</td>
<td>-0.31</td>
<td>-0.45</td>
<td>0.89</td>
<td>0.49</td>
<td>-0.41</td>
<td>-1.18</td>
<td>0.44</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.12</td>
<td>0.75</td>
<td>-0.43</td>
<td>-1.62</td>
<td>-0.30</td>
<td>0.03</td>
<td>-0.40</td>
<td>-1.54</td>
<td>-0.38</td>
<td>-0.11</td>
<td>-0.38</td>
<td>-1.55</td>
<td>0.00</td>
</tr>
<tr>
<td>0.09</td>
<td>-0.93</td>
<td>-1.22</td>
<td>0.47</td>
<td>-0.39</td>
<td>-0.29</td>
<td>-1.59</td>
<td>0.20</td>
<td>-0.42</td>
<td>-0.35</td>
<td>-1.05</td>
<td>0.18</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.51</td>
<td>-0.48</td>
<td>-0.59</td>
<td>-0.70</td>
<td>-0.13</td>
<td>-0.29</td>
<td>-0.84</td>
<td>-0.88</td>
<td>-0.08</td>
<td>-0.36</td>
<td>-0.69</td>
<td>-0.93</td>
<td>0.00</td>
</tr>
<tr>
<td>0.92</td>
<td>0.44</td>
<td>-0.46</td>
<td>-0.25</td>
<td>-0.38</td>
<td>0.03</td>
<td>-1.26</td>
<td>-0.67</td>
<td>-0.39</td>
<td>-0.11</td>
<td>-0.85</td>
<td>-0.63</td>
<td>0.00</td>
</tr>
<tr>
<td>0.09</td>
<td>1.08</td>
<td>0.26</td>
<td>0.46</td>
<td>-0.24</td>
<td>0.00</td>
<td>-0.15</td>
<td>0.34</td>
<td>-0.20</td>
<td>-0.24</td>
<td>-0.47</td>
<td>0.14</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.28</td>
<td>1.08</td>
<td>-1.78</td>
<td>-0.46</td>
<td>-0.35</td>
<td>3.00</td>
<td>-0.45</td>
<td>-2.24</td>
<td>-0.48</td>
<td>0.11</td>
<td>-1.09</td>
<td>-0.58</td>
<td>0.00</td>
</tr>
<tr>
<td>-0.73</td>
<td>1.49</td>
<td>-0.49</td>
<td>-1.11</td>
<td>-0.26</td>
<td>0.27</td>
<td>-0.52</td>
<td>-1.12</td>
<td>-0.42</td>
<td>0.32</td>
<td>-0.38</td>
<td>-1.04</td>
<td>0.00</td>
</tr>
<tr>
<td>0.40</td>
<td>0.33</td>
<td>-0.20</td>
<td>-0.51</td>
<td>-0.41</td>
<td>0.03</td>
<td>-0.20</td>
<td>-0.29</td>
<td>-0.45</td>
<td>0.02</td>
<td>-0.54</td>
<td>-0.73</td>
<td>0.00</td>
</tr>
<tr>
<td>2.01</td>
<td>0.83</td>
<td>-3.00</td>
<td>0.30</td>
<td>-0.35</td>
<td>0.14</td>
<td>-1.45</td>
<td>0.55</td>
<td>-0.40</td>
<td>0.00</td>
<td>-0.74</td>
<td>0.61</td>
<td>0.00</td>
</tr>
<tr>
<td>0.40</td>
<td>0.56</td>
<td>-0.59</td>
<td>0.08</td>
<td>-0.25</td>
<td>-0.14</td>
<td>-0.91</td>
<td>-0.10</td>
<td>-0.39</td>
<td>-0.15</td>
<td>-0.86</td>
<td>-0.25</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Generating Data Matrices from Data

Raw data

Array scans

Quantitation matrices

Quantitations

Gene expression data matrix

Genes

Samples

Spots

Gene expression levels
R, S and S-plus

S: an interactive environment for data analysis and a statistical programming language developed since 1976 primarily by John Chambers

Exclusively licensed by AT&T/Lucent to Insightful Corporation, Seattle WA. Product name: “S-plus”.

R: initially written by Ross Ihaka and Robert Gentleman during 1990s.

Since 1997: international “R-core” team of ca. 15 people with access to common CVS archive.

GNU General Public License (GPL), Open Source
What R does and does not

- data handling and storage: numeric, textual
- matrix algebra
- hash tables and regular expressions
- high-level data analytic and statistical functions
- classes (“OO”)
- graphics
- programming language: loops, branching, subroutines

- is not a database, but connects to DBMSs
- has no graphical user interfaces, but connects to Java, TclTk
- language interpreter can be very slow, but allows to call own C/C++ code
- no spreadsheet view of data, but connects to Excel/MsOffice
- no professional / commercial support
R and statistics

- Packaging: a crucial infrastructure to efficiently produce, load and keep consistent software libraries from (many) different sources / authors

- Statistics: most packages deal with statistics and data analysis

- State of the art: many statistical researchers provide their methods as R packages
S Language Elements

- Variables
- Missing values
- Functions and operators
- Vectors and arrays
- Lists
- Data frames
- Programming: branching, looping, subroutines
- apply
Vectors, matrices and arrays

**vector**: an ordered collection of data of the same type

```r
> a = c(1,2,3)
> a*2
[1] 2 4 6
```

**Example**: the mean spot intensities of all 15488 spots on a chip: a vector of 15488 numbers

**matrix**: a rectangular table of data of the same type

**Example**: the expression values for 10000 genes for 30 tissue biopsies: a matrix with 10000 rows and 30 columns.

**array**: 3-,4-,…dimensional matrix

**Example**: the red and green foreground and background values for 20000 spots on 120 chips: a 4 x 20000 x 120 (3D) array.
Data Frames Store Clinical/Biological Data Sets

data frame: is supposed to represent the typical data table that researchers come up with – like a spreadsheet.

It is a rectangular table with rows and columns; data within each column has the same type (e.g. number, text, logical), but different columns may have different types.

Example:

```
> a

  localization      tumorsize progress
XX348     proximal  6.3      FALSE
XX234     distal    8.0      TRUE
XX987     proximal  10.0     FALSE
```
apply

apply( array, margin, function )
Applies the function function along some dimensions of the array array, according to margin, and returns a vector or array of the appropriate size.

> x


> apply(x, 1, sum)
[1] 12 24 17 14

> apply(x, 2, sum)
[1] 22 25 20
Data Frame Example
(not real patient data)

"alk phos"
[16] 140 157 228 248 415 954 594 733 834 1785 3124 3582 3820 3459 3223
[31] 2259 2549 2111 1652 1098 1057 1098 1219 1803 1592 1525  943 1340 3268 4614
[46] 5900

$alt
[1] 26 63 360 141 179 44 28 21 27 22 19 19 14 17 18 27 22

"JHU Hb"
[1] 14.6 10.0 10.3 11.3 14.1 12.9 11.8 10.3 10.8 10.4 9.5 9.7 9.5 9.1 8.4
[16] 7.5 8.6 8.6 7.0 5.9 7.8 8.7 10.2 8.1 7.9 11.1 10.9 11.8 12.1 12.9
[31] 12.6 12.3 11.7 12.3 11.7 12.6 13.1 13.1 11.4  9.6 10.0  7.6  7.1  8.0  9.3
[46]  8.8

"JHU ICE COMBO"
[1] NA NA NA

"neut absolute"
[1]  2.250  1.030  1.680  0.983  0.740  0.854  0.981  0.785  1.060  0.857  0.570  3.600
[13]  2.690  2.900  1.100  1.100

$platelet
[20]  25  22  57 179  31  85 171 211 112 156 131 137 110 100  86 100 112 157 141
[39] 125 105  86  84  73  30  30  13  26  22
Ontologies for Events and Time Intervals

- Temporal Description Logic\(^2\)
  - 13 basic temporal interval relations (Allen notation)

<table>
<thead>
<tr>
<th>Relation</th>
<th>Abbr.</th>
<th>Inverse</th>
<th>i</th>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>before((i, j))</td>
<td>b</td>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meets((i, j))</td>
<td>m</td>
<td>mi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlaps((i, j))</td>
<td>o</td>
<td>oi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>starts((i, j))</td>
<td>s</td>
<td>si</td>
<td></td>
<td></td>
</tr>
<tr>
<td>during((i, j))</td>
<td>d</td>
<td>di</td>
<td></td>
<td></td>
</tr>
<tr>
<td>finishes((i, j))</td>
<td>f</td>
<td>fi</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: Concept “Clinical Type II Rejection”

- Type-II-Rejection:
  OVERLAPS(Bili_Fever, 
  UNION( Int(“GOT=increase”), 
  Int(“GPT=increase”)), 
  “days”) AND 
OVERLAPS([4,21], Bili_Fever, “days”) 
RESULT: 
Start(Bili_Fever),Finish(Bili_Fever)

- Bili_Fever: 
DURING(Int(“MaxTemp=Fever”), 
  High_Bili_Increase, “days”) 
RESULT: 
Start(High_Bili_Increase),Finish(High_Bili_Increase)

- High_Bili_Increase: 
During(Int(“Bilirubin=high”), 
  Int(“Bilirubin=increase”),”days”) 
RESULT: 
Start(Bili_Increase),Finish(Bili_Increase)

Retrieve all occurrences of patient episodes, where the interval representing increase of bilirubin with at least partly fever episodes overlaps an interval representing an increase of transaminases (GOT or GPT) within day 4 and day 21 after liver transplantation.

This concept is characterized by the interval of bilirubin increase.

The concept bili_increase represents occurrences with values at least partially over 100 umol/l
Concept
categorical entity of the domain

Property
attribute describing a concept

Relation
relationship between concepts or properties

Axiom
coherecncy description between Concepts / Properties / Relations via logical expressions

\[ \text{hasAdmissionDate}(\text{Patient}) \implies \text{admitted}(\text{Patient}) \]
High and Increasing Bilirubin

Patient(?p)
hasProcedure(?p, ?proc)
  hasTest(?proc, ?test)
  hasTestName(?test, ?testName)
  swrlb:equal(?testName, "BILIRUBIN")
  HasOutputType(?test, ?testType)
  swrlb:equal(?testType, "INCREASE")
  temporal:hasValidTime(?test, ?tVT)
  hasTest(?proc, ?test2)
  hasTestName(?test2, ?testName2)
  swrlb:equal(?testName2, "BILIRUBIN")
  HasOutputType(?test2, ?testType2)
  swrlb:equal(?testType2, "HIGH")
  temporal:hasValidTime(?test2, ?tVT2)
    temporal:overlaps(?tVT, ?tVT2, "days")
  temporal:hasStartTime(?tVT, ?stTime)
  temporal:hasFinishTime(?tVT, ?fiTime)
swrlx:createOWLThing(?hbVT, ?proc) ->temporal:ValidPeriod(?hbVT)
  temporal:hasStartTime(?hbVT, ?stTime)
  temporal:hasFinishTime(?hbVT, ?fiTime)
  hasHighBiliIncrease(?proc, ?hbVT)
Discussion

- Proof of concept
- SPOT is a feasible approach to use open source and standards based software
- Different solutions to “translate” logic from OWL/SWRL into S
- Currently, concept intervals are passed from OWL/SWRL through the Java interface and “relearned” through a classification tool in R, e.g., discriminant analysis.
- SWRL interface improved with modularization since object instantiation is possible
- Need of GUI for researcher
Acknowledgements

Thank you

- Mark Musen
- Tania Tudorache
- Samson Tu
- Ted Hopper
- The Protégé Team at Stanford
Thank you for your attention