Ranking episodes using a partition model

Nikolaj, Tatti@aalto.fi
Aalto University, Helsinki Institute of Information Technology

Ranking episodes

Episodes are
(i) patterns occurring in sequences,
(ii) order of events is described by DAGs,
(iii) gap events are allowed.

Example:
Episode $G$ occurs in a sequence if and only if (i) $a$ occurs, (ii) then $b$ and $c$, in any order, (iii) and then $d$. Gap events are allowed.

For a set of sequences $S$, the support is

$$supp(G) = \{S \in S \mid G \text{ occurs in } S\}.$$ 

Mining using support counter-productive:
(i) output is humongous,
(ii) pattern are redundant

Rank patterns based on expectation.
Computing expectation is more intricate than with itemsets:
(i) models are difficult to compute,
(ii) depends on the sequence length

Assume we can get

$$p(n) = p(G \text{ occurs in a sequence of length } n)$$

Then the expected support is

$$\mu = \sum_{S \in S} p(|S|).$$

Compare $supp(G)$ and $\mu$:
The larger the difference, the more important is the episode.

Independence model

We need

$$p(\text{episode occurs in a sequence of length } n)$$

No closed form but can be computed.

Example:
Episode $G$ occurs in a sequence if and only if we reach $H_6$ from $H_1$ in $M$.

To compute the probability use

$$p(H, n) = q \times p(H, n-1) + \sum_{e \in (F,H) \in E(M)} p(e)p(F, n-1),$$

where $q$ is the probability of being stuck in $H$ for a single event

$$q = 1 - \sum_{e \in E(M)} p(e).$$

Free-rider episodes

Independence model does not get rid of free-riders.

Example: If $G$ is significant, then $G'$ is also significant even if $x$ is independent of $G$.

Partition model

General idea:
(i) split the episode into two subepisodes,
(ii) study how often these episodes occur,
(iii) incorporate this into a model,
(iv) try all splits, and use the best.

Example: Split $G$ into $G_1$ and $G_2$

Model how likely $G_1$ is discovered once we see at least one event in $G_i$.

In practice, boost the probabilities $p(e)$ in

$$p(e) \propto \begin{cases} \exp(u_i + t_1), & \text{if } e \in C_1, \\ \exp(u_i + t_2), & \text{if } e \in C_2, \\ u_i, & \text{otherwise}. \end{cases}$$

Parameters $u_i$ and $t_i$ can be learned by maximizing likelihood; gradient descent will converge to the global optimum.

Superepisodes

Similar to partition model except now test for superepisodes.

Example: $F$ is a superepisode of $G$

Model how likely $F$ is discovered once we see at least one event in $F$.

In practice, boost the probabilities $p(e)$ in

Episodes with high $r_{ind}$ and low $r_{prt}$

$G_1$: support $\rightarrow$ vector $\rightarrow$ machin $\rightarrow$ regress $95.1$ $\rightarrow$ regress
$G_2$: support $\rightarrow$ vector $\rightarrow$ machin $\rightarrow$ regress $90.4$
$G_3$: support $\rightarrow$ vector $\rightarrow$ machin $\rightarrow$ number $52.0$ $\rightarrow$ regress
$G_4$: support $\rightarrow$ vector $\rightarrow$ machin $\rightarrow$ space $51.6$

Experiments

Top episodes in Plant dataset. The symbols $x$ and $y$ represent noise events.

<table>
<thead>
<tr>
<th>Independence model</th>
<th>Partition model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>Episode type</td>
</tr>
<tr>
<td>1.</td>
<td>$a \rightarrow b \rightarrow c \rightarrow d$</td>
</tr>
<tr>
<td>2.</td>
<td>$k \leftarrow m \rightarrow l \rightarrow i$</td>
</tr>
<tr>
<td>3.</td>
<td>$a \rightarrow b \rightarrow d \rightarrow x$</td>
</tr>
<tr>
<td>4.</td>
<td>$c \leftarrow f$</td>
</tr>
<tr>
<td>9.</td>
<td>$x \rightarrow y$ or $x, y$</td>
</tr>
</tbody>
</table>

Episodes discovered from JMLR abstracts:

<table>
<thead>
<tr>
<th>ranked by $r_{ind}(G)$</th>
<th>$r_{ind}$</th>
<th>$r_{prt}$</th>
<th>ranked by $r_{prt}(G)$</th>
<th>$r_{ind}$</th>
<th>$r_{prt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>support $\rightarrow$ vector $\rightarrow$ machin</td>
<td>$\infty$</td>
<td>357</td>
<td>support $\rightarrow$ vector</td>
<td>440</td>
</tr>
<tr>
<td>2.</td>
<td>support $\rightarrow$ vector</td>
<td>440</td>
<td>440</td>
<td>support $\rightarrow$ vector $\rightarrow$ machin</td>
<td>$\infty$</td>
</tr>
<tr>
<td>3.</td>
<td>support $\rightarrow$ vector $\rightarrow$ machin $\rightarrow$ svm</td>
<td>404</td>
<td>90</td>
<td>support $\rightarrow$ machin</td>
<td>324</td>
</tr>
<tr>
<td>4.</td>
<td>support $\rightarrow$ vector $\rightarrow$ machin $\rightarrow$ svm</td>
<td>$10^{-5}$</td>
<td>vector $\rightarrow$ machin</td>
<td>306</td>
<td>306</td>
</tr>
<tr>
<td>5.</td>
<td>reproducability $\rightarrow$ kernel $\rightarrow$ hilbert $\rightarrow$ space</td>
<td>341</td>
<td>73</td>
<td>data $\rightarrow$ set</td>
<td>284</td>
</tr>
<tr>
<td>6.</td>
<td>support $\rightarrow$ machin</td>
<td>325</td>
<td>325</td>
<td>real $\rightarrow$ world</td>
<td>260</td>
</tr>
<tr>
<td>7.</td>
<td>vector $\rightarrow$ machin</td>
<td>306</td>
<td>306</td>
<td>real $\rightarrow$ data</td>
<td>213</td>
</tr>
<tr>
<td>8.</td>
<td>data $\rightarrow$ set</td>
<td>284</td>
<td>284</td>
<td>state $\rightarrow$ art</td>
<td>191</td>
</tr>
<tr>
<td>9.</td>
<td>real $\rightarrow$ world</td>
<td>260</td>
<td>260</td>
<td>machin $\rightarrow$ learn</td>
<td>190</td>
</tr>
<tr>
<td>10.</td>
<td>support $\rightarrow$ vector $\rightarrow$ svm</td>
<td>250</td>
<td>85</td>
<td>bayesian $\rightarrow$ network</td>
<td>166</td>
</tr>
</tbody>
</table>