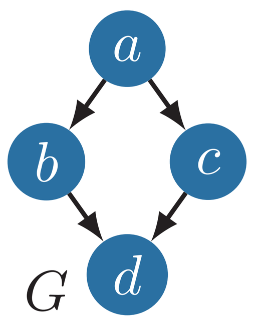
RANKING EPISODES USING A PARTITION MODEL

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 $\mathcal{S}$, the support is

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

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

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

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

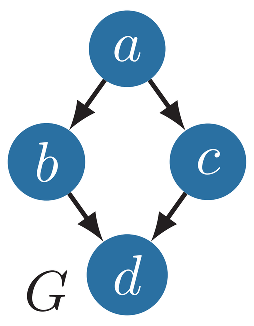
INDEPENDENCE MODEL

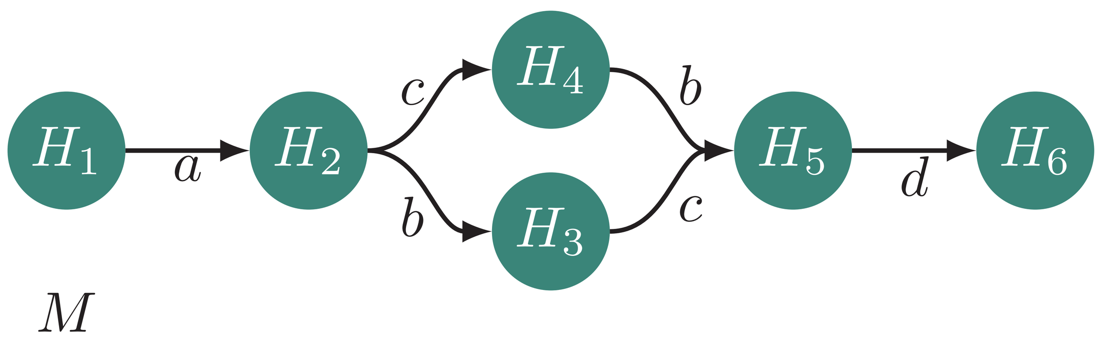
We need

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

No closed form but can be computed.

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





To compute the probability use

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

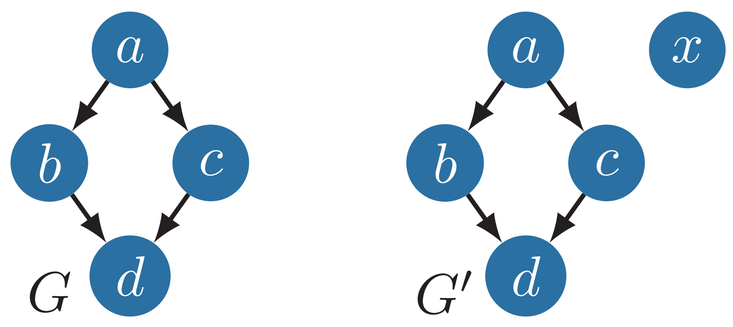
where q is the probability of being stuck in H for a single event

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

FREE-RIDER EPISODES

Independence model does not get rid of free-riders.

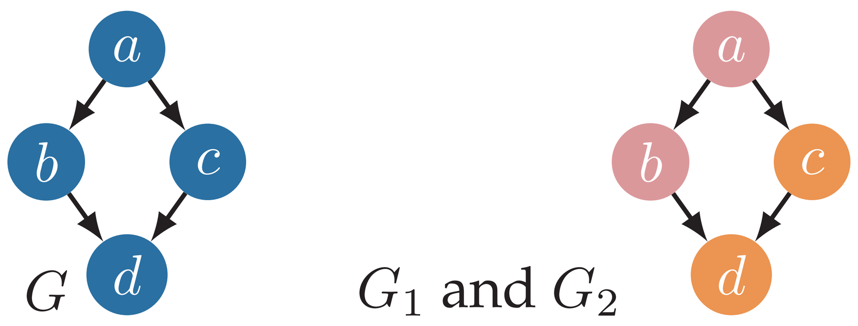
Example: If G is significant, then G' is also significant even if x is independent of G.



PARTITION MODEL

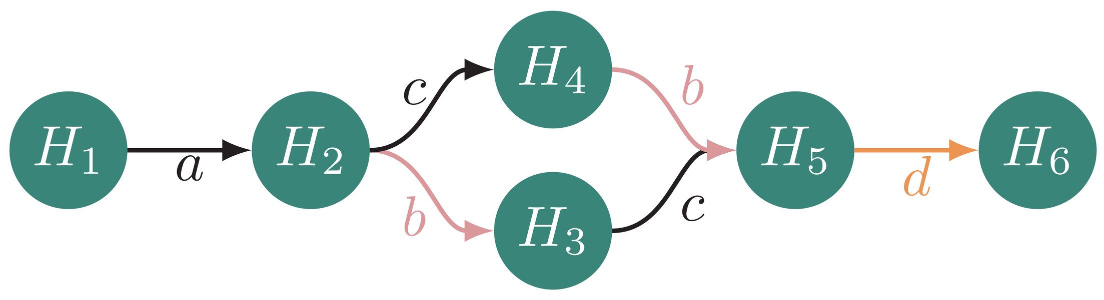
General idea:
(i) split the episode into two subepisodes,
(ii) study how often these episodes occur,
(iii) incorporate this into a model,
(iv) try all splits, and use the best.

Example: Split G into G_1 and G_2



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

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



Boosted probabilities leads to higher expected support, and lowered rank.

Edge e with label l is modeled as

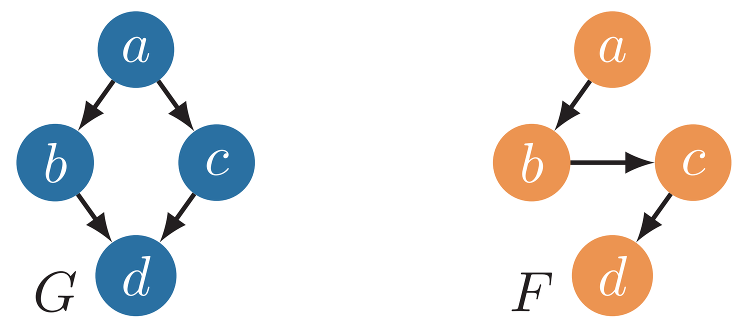
$$p(e) \propto \begin{cases} \exp(u_l + t_1), & \text{if } e \in C_1, \\ \exp(u_l + t_2), & \text{if } e \in C_2, \\ \exp(u_l), & \text{otherwise} \end{cases} \quad .$$

Parameters u_l and t_1 can be learned by maximizing likelihood; gradient descent will converge to the global optimum.

SUPEREPISODES

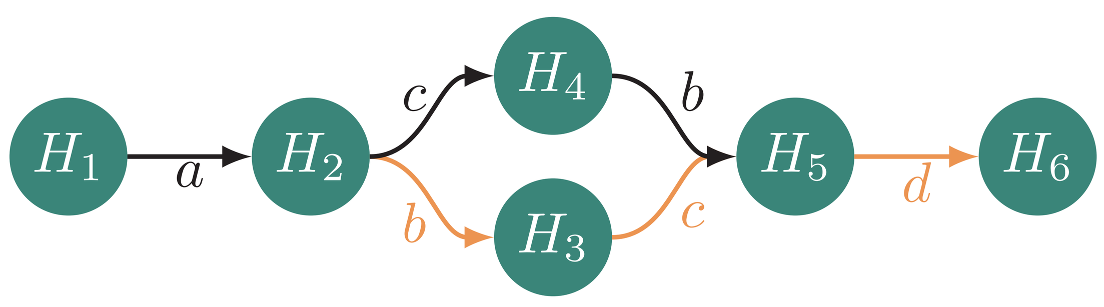
Similar to partition model except now test for superepisodes.

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



EXPERIMENTS

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

Independence model			Partition model		
Rank	Episode type	$r_{ind}(G)$	Rank	Episode type	$r_{prt}(G)$
1.	$a \to b \to c \to d$	∞	1.	$a \to b \to c \to d$	10^{308}
2.	$k \to n, k \to m, n \to l, m \to l$	249	2.	$e \to f$	128
3.–7.	$a \to b \to c \to d \to x$	184–185	3.	$k \to n, k \to m, n \to l, m \to l$	78
8.	$e \to f$	128	4.–	$a \to b \to c \to d \to x$	0–14
9.–	$x \to y$ or x, y	2–14		or $x \to y$ or x, y	

Episodes discovered from *JMLR* abstracts:

	ranked by $r_{ind}(G)$	r_{ind}	r_{prt}	ranked by $r_{prt}(G)$	r_{ind}	r_{prt}
1.	support→vector→machin	∞	357	support→vector	440	440
2.	support→vector	440	440	support→vector→machin	∞	357
3.	support→vector→machin→svm	404	90	support→machin	324	324
4.	support→vector→machin svm	356	10^{-3}	vector→machin	306	306
5.	reproduc→kernel→hilbert→space	341	73	data→set	284	284
6.	support→machin	325	325	real→world	260	260
7.	vector→machin	306	306	real→data	213	213
8.	data→set	284	284	state→art	191	191
9.	real→world	260	260	machin→learn	190	190
10.	support→vector→svm	250	85	bayesian→network	166	166

Episodes with high r_{ind} and low r_{prt}
G_1: support → vector → machin regress   95.1
G_2: support→vector→machin, support→regress   90.4
G_3: support → vector → machin number   52.0
G_4: support→vector→machin, vector→regress   86.4
G_5: support → vector → machin space   51.6