Mining Episodic Memory

Nate Derbinsky

Research Statement

Analyze the contents of episodic memory across a set of tasks/agents for useful patterns

<u>Goals</u>

- Inform architectural additions/enhancements
- Inform agent design

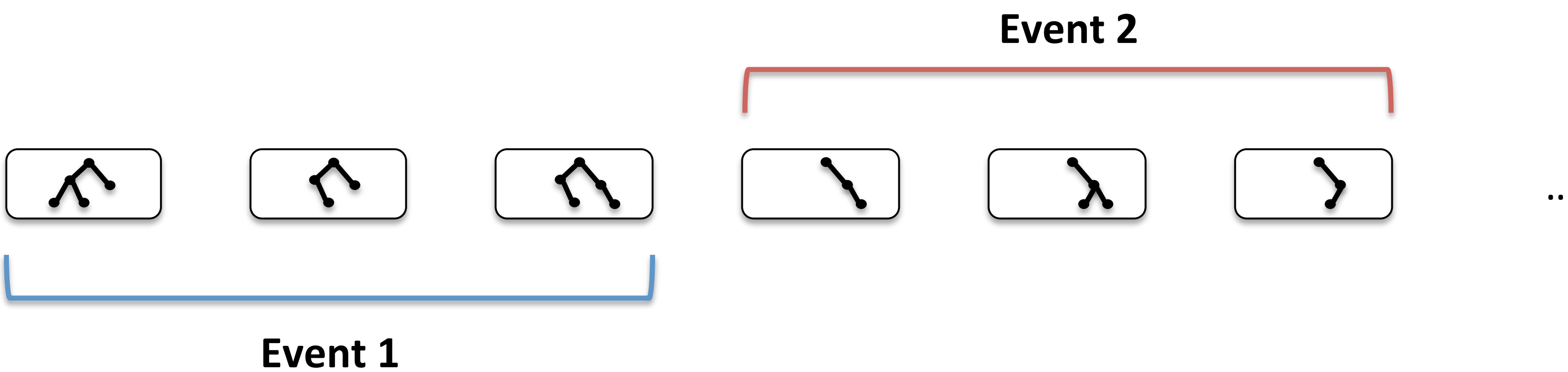
<u>Initial Directions</u>*

- Episodic summarization & forgetting
- Episodic-to-semantic consolidation

* *Offline, ignoring issues of performance*

Direction 1

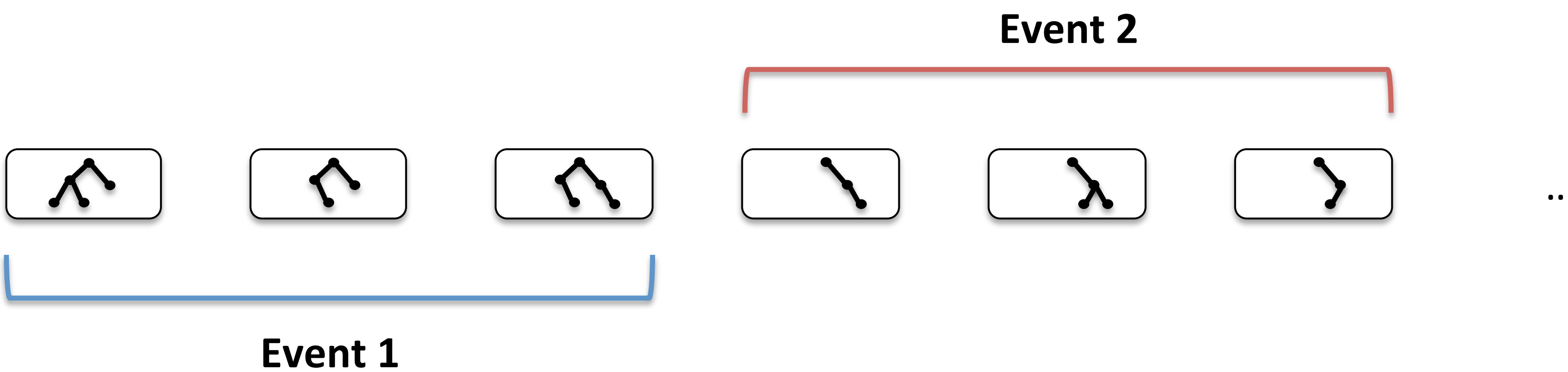
- Basic idea: event recognition
  - Analyze the WME stream and identify groupings that correspond, along some metric, to distinct "events”



Episodic Summarization/Forgetting

Over time, temporally local episodes reduce in specificity

- Coalesce into representative "events”
- Statistical noise pruned/summarized



Potential Outcomes

- Improved EpMem performance
  - Fewer episodes = stabilized query time
  - *Hopefully* equal/better quality
    - Hypothesis: detail useful only in short-term
- Comparison to human cognition
  - Literature: "episodic memories are encoded without great detail”
    - Proposal: progression from high-to-low detail

Candidate Metrics/Signals

- Mechanism performance
  - More WME changes = slower EpMem
  - Serves as a pressure to prune where possible
- Feature/episode...
  - Similarity ala graph structure, temporal distance, feature-level variance (~TF/IDF), latent events (~LSA)
  - Activation
    - Episode: storage recency, retrieval base-level
- Agent queries
  - Cue = possible signal of details worth keeping
- **Goal**. Content-aware compression

Initial Experimental Framework

1. Gather multi-task episodic databases
   – Sanity check: big switch across unrelated in-the-head agents (counting, water-jug, etc.)

2. Attempt to recover task sequence
   – *Unsure how to evaluate summarization for non-trivial agents*

Candidate Model #1

kMeans in WM space

- Features ala Working Memory Tree
  - Flattens multi-valued attributes, for simplicity
  - Binary feature vector (*bag of words*)
- Greedy search for “optimal” k
  - Distance: {hamming, cosine, hybrid?}
  - Smaller k
    - **Greater compression**
    - Greater within-event variance
    - Greater exemplar variance
  - Larger k
    - Less compression
    - **Less within-event variance**
    - Less exemplar variance

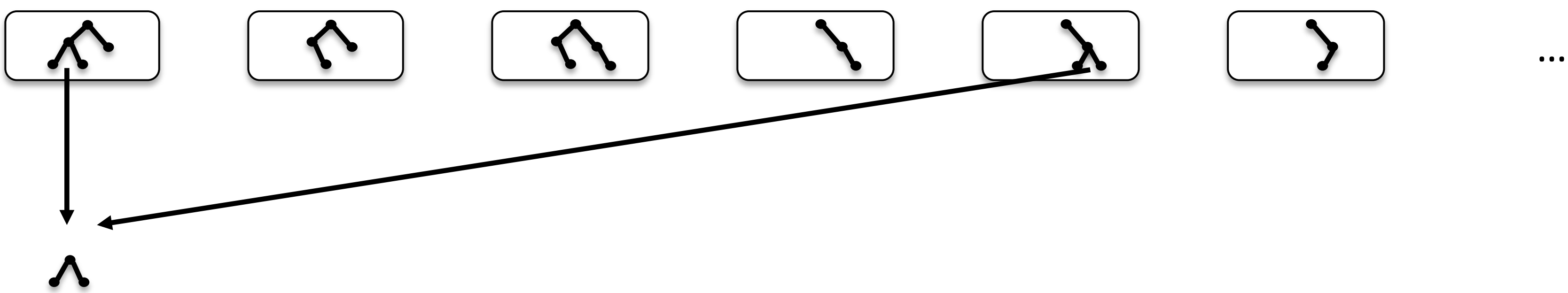
Candidate Model #2

Singular Value Decomposition (SVD)

- Identify features/feature sets that explain the most episode variance
  - Consider value changes = event boundaries
- Smooth within-event feature changes for features that explain little episode variance

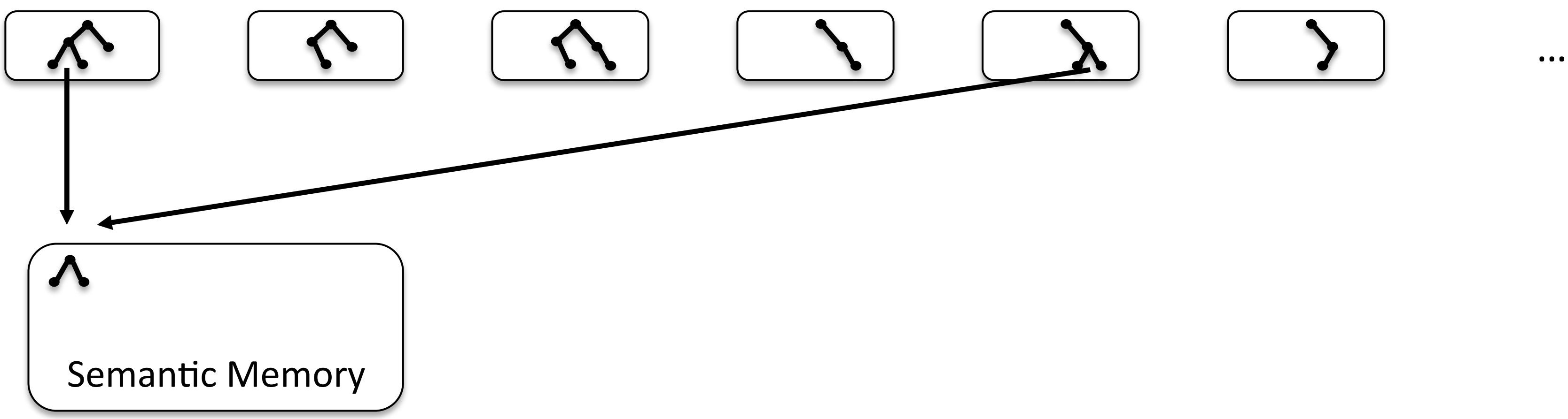
Direction 2

- Basic idea: online concept learning
  - Analyze the WME stream and identify groupings of WMEs that, over time, capture stable distributions of object features, independent of context



Potential Outcomes

- Architectural semantic storage!
  - Useful categorical hierarchies/clusters
  - Summarization of probabilistic distributions



Classes of Candidate Features

1. Individual, over time
   - Variance (e.g. TF/IDF, density estimation)
   - Type (numerical/categorical)
   - Structural location (e.g. input-link, MVAs)

2. Pairwise relations over time
   - Co-variance
   - Structure mapping [w.r.t. existing LTIs]

3. Architectural meta-data
   - Episodic/semantic cues
   - RL reward/update error
   - SVS filtering

Candidate Models

1. Feature co-occurrence matrix
   – Mine for single-level (ala Working-Memory Tree) frequent item sets (e.g. Apriori)
     - Feature variance -> keep vs. discard vs. parameters (e.g. mean/variance)

2. Incremental COBWEB/TRESTLE (MacLellan et al. ACS'15)

Initial Experimental Framework

Issue. What "should" a learned semantic network look like for a domain?

– *I don't have a great answer here…*

1. Synthetic dataset with known priors (e.g. GMM), try to recover
   - Wang: hunting domain
     - Priors on weapon effectiveness w.r.t. prey type
   - Map construction

Potential Interactions

- Episodic category learning
  - In addition to learning concepts, it can be useful to be able to categorize types of events (Doshi, Kira, Wagner ACS'15)
  - Example application: **spontaneous retrieval** of episodes
    - Train a situational classifier, automatically retrieve/signal availability of a similar episode/ category exemplar
- Spontaneous retrieval as recommender system
  - Given prior requests, the current context, and **known co-occurrences**, suggest an LTI to supplement the existing context
  - Form of hybrid content/collaborative filtering
- Episodic compression via LTI re-encoding
  - When an episode is retrieved, re-encode subsets of the graph that were STIs with LTIs that have **since been learned**
  - Has the potential to speed/improve retrieval
- Elaborative chunk learning
  - When a concept is learned from perceptual data, automatically augment symbolic interpretation with **learned conceptual LTI**
  - Over time, re-use to learn higher-level features

Evaluation

Nuggets

- Finally looking to episodic memory as a source of useful statistical data
  - Hopefully full of
- Initial directions lay the groundwork for future architectural mechanisms
  - Episodic summarization, automatic semantic storage
- Potential for beneficial interactions

Coal

- No results yet
- Huge space of potential signals, metrics, approaches; no standard

Thank You :)

Questions?

Comments?

Suggestions?

Concerns?