DISCOVERY OF NWC PATTERNS

Low-level Design

Classes

EngineReading

(all data except Q-dimensions)

PatternDimension

(name, index, min, max, intervalwidth, **intervals**)

OutputPattern

(pattern, patterncount, pa_count, distinct_pa_count, kvalue, anomalycount)

HybridDimensionNode

(dimensions, isPruned, superpattern_count, parents_num, children_num, **children**, **parents**)

TemporalWindow

(startIteration, endIteration)

EventInterval

(start, end)

AllStatesGraph

(dimensionsNum, **statesGraph**)

HybridDimensionGraph

(leaves_pa_count, pa_count, leaves_pattern_count, distinct_pa_count, topLevel, bottomLevel, **graphNodes**)

Data Structures

allReadings
<EngineReading>

Intervals
<EventInterval>

PatternDimensions
<PatternDimension>

topKFunctionValuePatterns
<OutputPattern>

graphNodes
<<HybridDimensionNode>>

children / parents
<HybridDimensionNode>

stateGraphs
<{String, <Integer>}>

includedDimensions
<Integer>

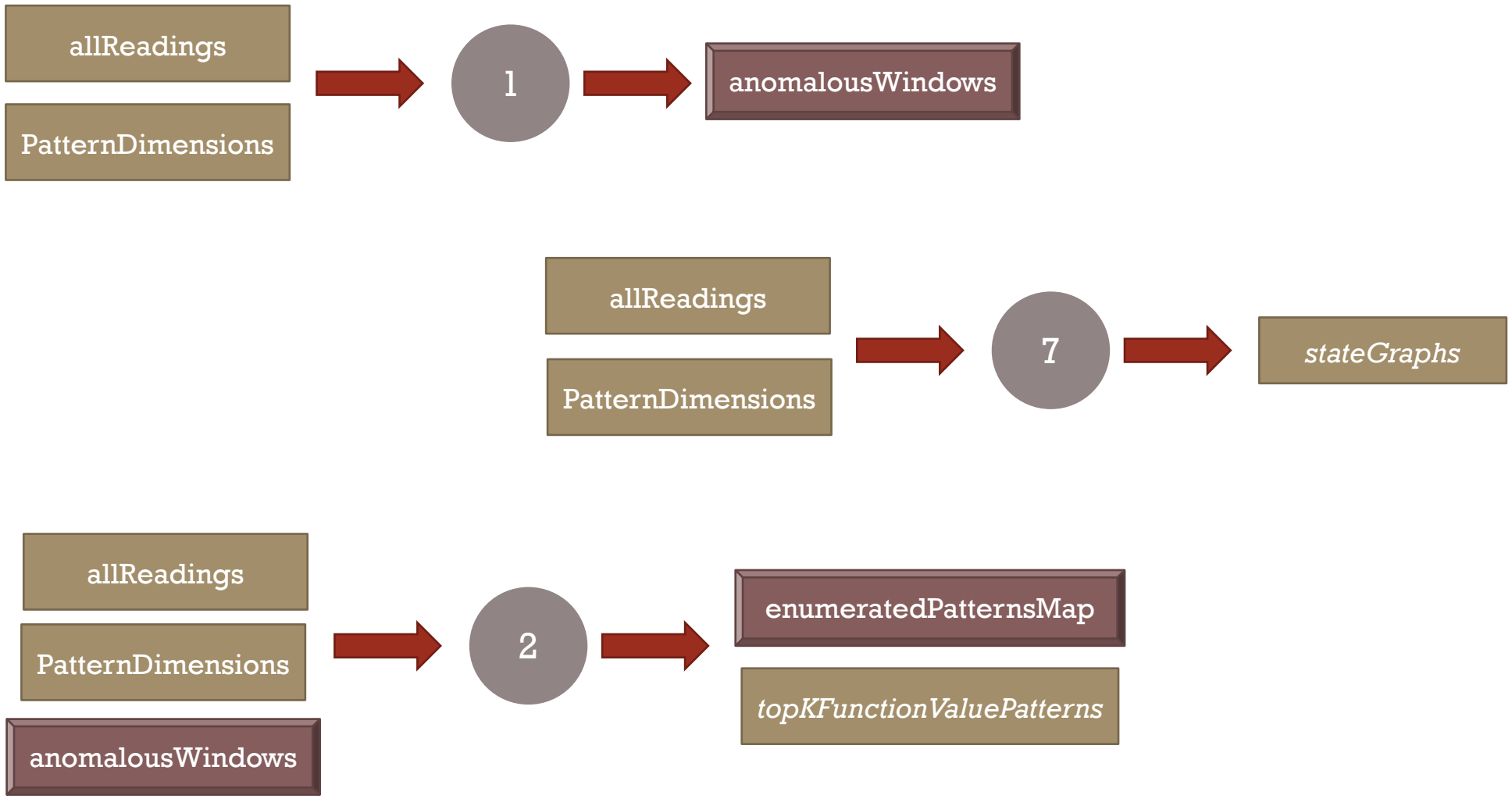
anomalousWindows
<String, TemporalWindow>

enumeratedPatternsMap
<String, String>

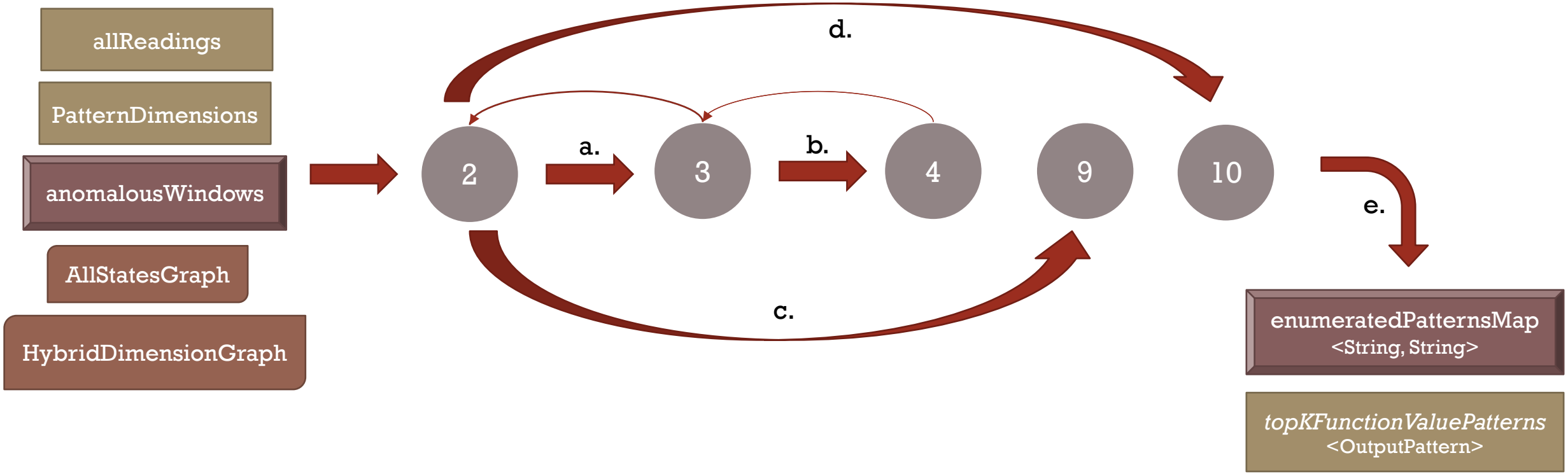
MODULES

1. findanomalousWindows()
2. mineCooccurrencePatterns()
3. enumerateSingleton()
4. expandSingletonPattern()
5. expandPattern()
6. pruneAllParents()
7. countPatternUsingStateGraphs()
8. calculateKFunctionAndOutputIfPatternComplies()
9. enumerate_with_UB_pruning()
10. enumerate_with_minsupp_pruning()
11. printAllSizeKSubsets()
12. pruneAllDescendants()
13. pruneDescendantsBecausePreviouslyEnumerated()

Control Flow



BDNMiner


allReadings
PatternDimensions
anomalousWindows
AllStatesGraph
HybridDimensionGraph
2
3
4
9
10
a.
b.
c.
d.
e.
enumeratedPatternsMap
<String, String>
topKFunctionValuePatterns
<OutputPattern>

HELPER DOCS

- **findanomalousWindows()**

Input: allReadings

Output: anomalousWindows

Description: Finds anomalous windows of specified length, with unexpected behavior and in same time and pass.

- **mineCooccurrencePatterns()**

Input: allReadings, anomalousWindows, thresholds, lag

Output: topKpatterns

Description: Mines and retrieves patterns co-occurrence patterns in all of data, using pruning based optimizations

- **enumerateSingleton()**

Input: HybridDimensionNode, windowFrom, windowTo, lag

Output: True / False of whether leaf node is pruned or not

Description: Counts & Enumerates leaves, if support < min ; then prunes parents ; else calculates cross_k

- **expandSingletonPattern()**

Input: allReadings, PatterDimensions, from, to

Output: Expanded pattern for leaves (String)

Description: Uses info and prepares key for leaves of Q-dimension as `name  from-to space separated values`

- **expandPattern()**

Input: allReadings, PatterDimensions, from, to, includedDimensions

Output: Expanded pattern for all included dimensions in the node (String)

Description: Uses info and prepares key for included Q-dimensions as `name1  values1 \n name2  values2 …`

- **pruneAllParents()**

Input: HybridDimensionNode, HybridDimensionGraph

Output: None

Description: Recursively propagates through all parent nodes of the given node and sets their isPruned flag to true

- **countPatternUsingStateGraphs()**

Input: allReadings, includedDimensions, anomalousWindows, windowFrom, windowTo, from, to, lag

Output: pattern_count, pattern_anomaly_count, distinct_pa_count (one anomaly considered per window match, no lag)

Description: First gets startTimelist of first included dimension, then compares all in included dimensions in from-to window to windows in startTimelist for loop. If pattern found in a startTimelist window, anomalous window is searched around it.

- **calculateKFunctionAndOutputIfPatternComplies()**

Input: pattern_counts, pattern_support, total_anomalous_windows

Output: None

Description: Calculates cross_k for pattern, and if above threshold, outputs pattern to file and inserts in Top-K patterns.

- **enumerate_with_UB_pruning()**

Input: allReadings, HybridDimensionNode & Graph, anomalousWindows, from, to, thresholds, stateGraphs

Output: None

Description: Top-down enumeration & pruning of nodes via UB_local and UB_lattice, passing supersetcount to children.

- **enumerate_with_minsupp_pruning()**

Input: allReadings, HybridDimensionNode & Graph, anomalousWindows, from, to, thresholds, stateGraphs

Output: True / False if the nodes was pruned or not

Description: Bottom-up enumeration & pruning of nodes via Apriori / support, storing JoinsetCounts for tight bounds calc.

- **printAllSizeKSubsets()**

Input: inputDimensions, empty subsets arrayList, nextBottomLevel

Output: None (adds comma separate combinations to subsets arraylist)

Description: Recursively generates all combinations of dimensions in the node, at the recently traversed K nextBottomLevel.

- **pruneAllDescendants()**

Input: HybridDimensionNode, HybridDimensionGraph

Output: None

Description: Called once in UB_pruning, recursively prunes descendants of given node.

- **pruneDescendantsBecausePreviouslyEnumerated()**

Input: HybridDimensionNode, HybridDimensionGraph

Output: None

Description: Same as the above function, just a different counter within