

Self Organising Map Approaches for clustering of XML documents

Ah Chung

ADMA06 Keynote Speech

August 14-16, Xi'an China

Contents

- Monash/Wollongong group
- Self organising map for structures
- XML documents
- Self organising map for structures applied to XML document clustering
- Conclusions

Monash/Wollongong group

- Web page ranking (personalised, web portal)
- Graph neural networks and applications
- Data mining of temporal data
- Large scientific databases and issues
- Blind source separation
- Adaptive neuro-fuzzy approaches
- Self organising map approach to clustering

Machine learning techniques

	Deterministic	Stochastic
Static	Feedforward NN Kernel machine	Mixture models
Sequential	Recurrent NN String kernel	HMM, IOHMM
Structural	Recursive NN Graph kernel	Recursive HMM

Self organising map approach

- Originated from Teuvo Kohonen
- A very popular method for clustering, dimension reduction, data reduction, data visualisation
- Works very simply and efficiently

Self organising map algorithm

- Architecture: two dimensional array of neurons
- Choose a vector from the training data set
- Compare this vector with the codebook vector that is stored in each neuron, and find the winning neuron (in terms of similarity)
- Update the codebook vectors
- Repeat for every vector from the training data set and for a prescribed number of times

Graph data structures

Practical data sometimes are more
conveniently modelled using graphs

Examples: Chemistry, Image, Document, ..

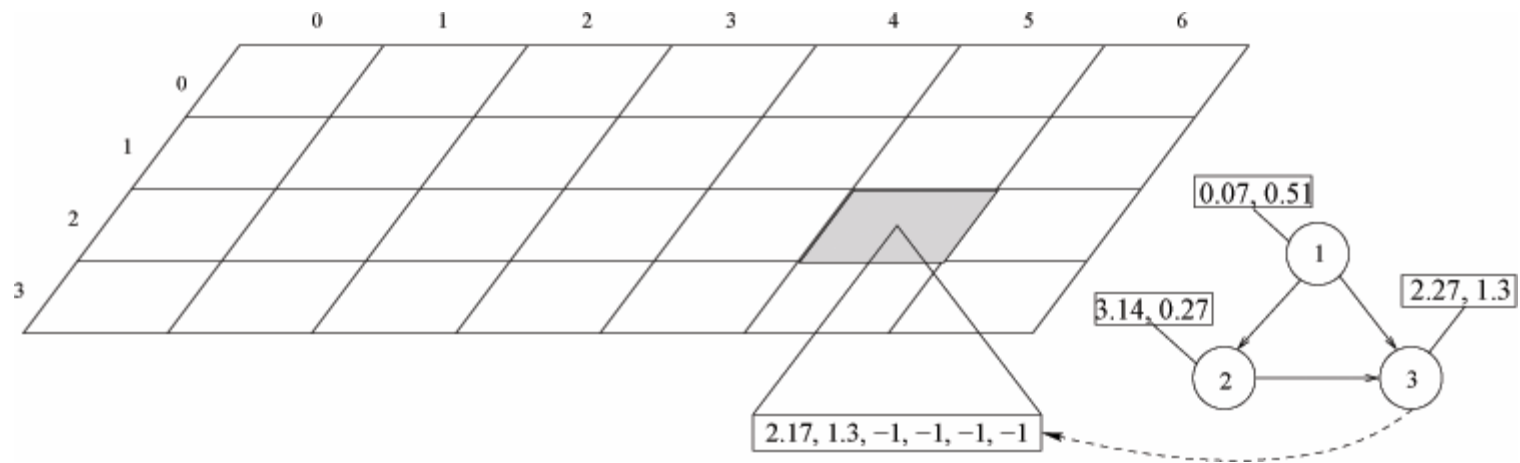
Approaches

- Two approaches to processing graph data
 - Flatten the graph structure so that it becomes a vector.
 - Preserve the structure of the data until it is necessary to process the data
- Unsupervised and supervised learning
- Will consider only unsupervised learning from now on

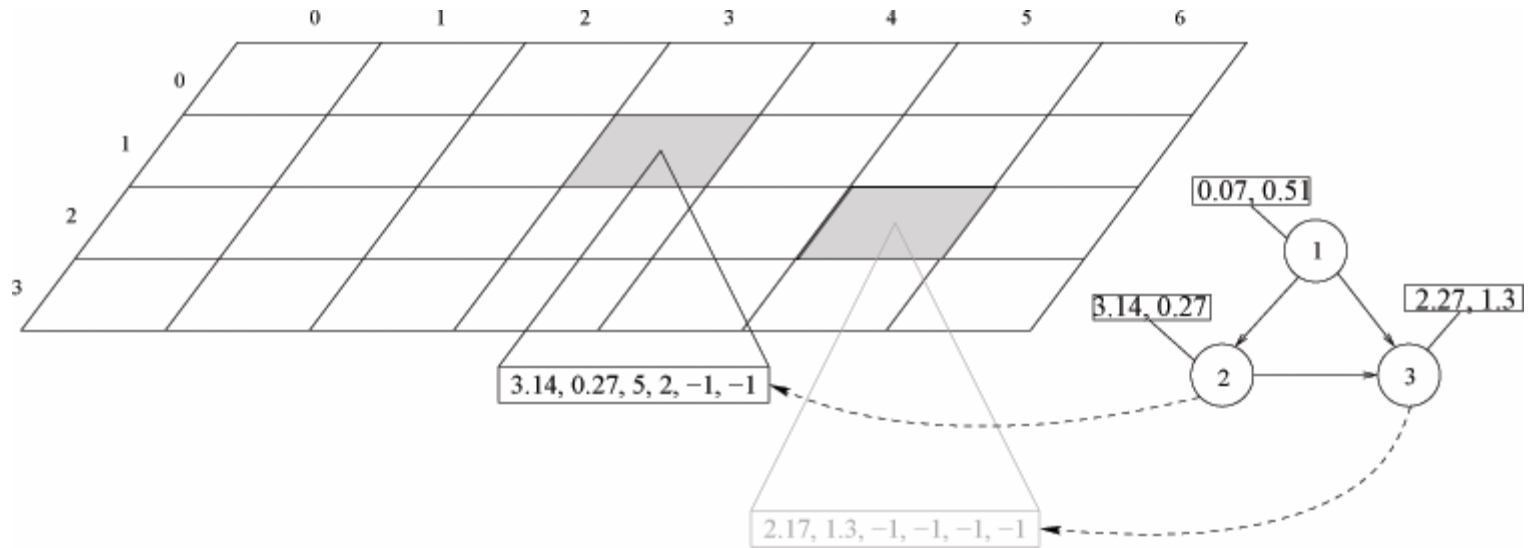
Structure processing

- Process one node at a time
- Obtain the state of the neighbours, e.g. the immediate descendent, with the winning neuron coordinates (neighbours here could include ancestors, and descendents)
- Concatenate node label with state information
- Process each vector as in standard SOM but with weighting on node label and state vector

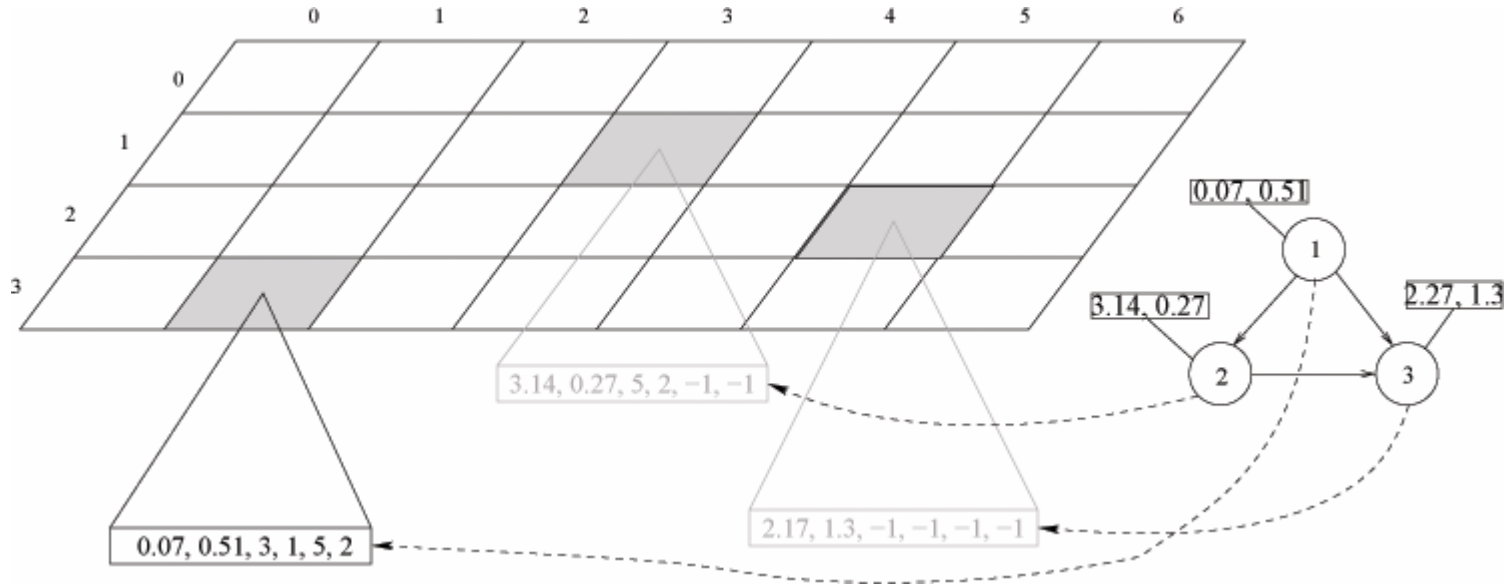
Example: Step 1



Example: Step 2



Example: Step 3



Algorithm

- Choose a node in the graph from the training set
- Form a vector $x=[\text{label}, \text{state info}]$
- Find the winning neuron by comparing this concatenated vector with the code vector in the SOM (weighted)
- Update the weight of the network
- Repeat for every node in the training set and for a prescribed number of times

Performance measure

- Retrieval capability (R)
- Classification performance (C)
- Clustering performance (P)
- Structural mapping precision (e)

XML documents

- Internet contains x billions of web pages of HTML/XML documents
- Important to develop automatic tools to analyse them – machine learning method
- Web pages are not labelled – unsupervised learning, clustering
- XML documents can be represented as a structured document – trees/graphs

XML

```
<AW>  
  <BC>  
  </BC>  
  <BG>  
  </BG>  
  <AH>  
    <DU>  
      <BJ>  
      </BJ>  
      <AJ>  
      </AJ>  
    </DU>  
  </AH>  
  <BT>  
  </BT>  
  <CG>  
  </CG>  
</AW>
```

INEX dataset

- M-db-s-0 dataset consists of 9,640 documents
- Only XML tags (197 different tags)
- All documents have targetted values (11 classes)
- Use 4,824 for training
- Max depth of tree:3; Max outdegree: 6,418
- Total number of nodes: 684,191

Preprocessing

- Consolidation

<BB>

<a>

<a>

</BB>

Becomes

<BB>

<a>

</BB>

Preprocessing

- Collapse

- `<A> <c> </c> ` becomes

- `<A> <b & c> </b & c> `

- `<A b & c> </A b & c>`

After preprocessing maximum outdegree is 32, and there are a total of 124,359 nodes

Characteristics of training set

Class	1	2	3	4	5	6	7	8	9	10	11
Freq	598	486	701	172	435	231	261	769	333	386	448

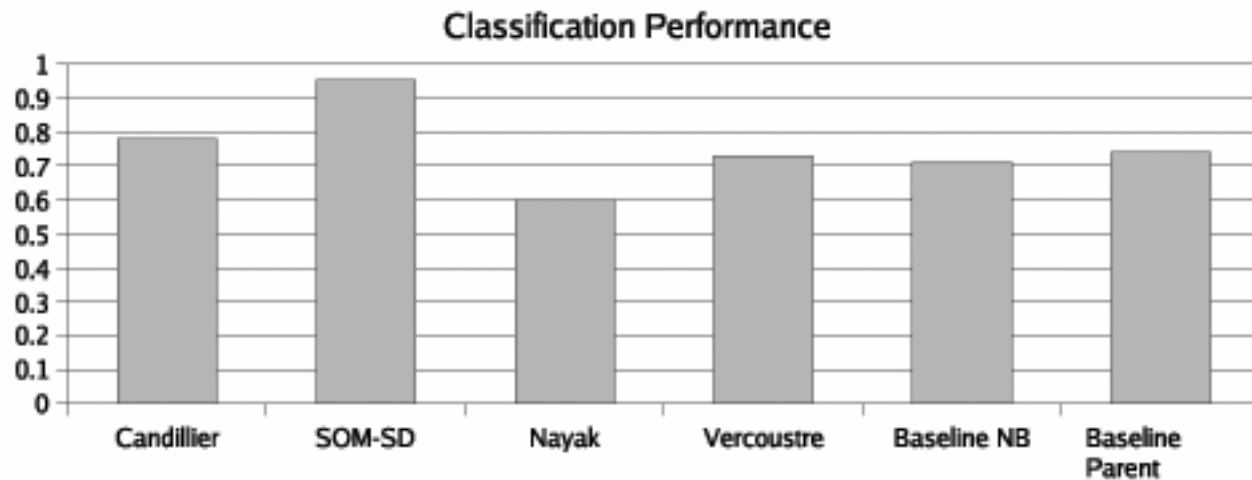
Experiments

- Flatten the XML documents into vectors of 197 long (being the total number of distinct tags). No structural information is retained.
- Run SOM-SD through the data with structures preserved (with preprocessing).
- Compare the results

Results: SOM (64x48); SOM-SD (89x67)

	Training					Testing			
	R	e	P	C	Z	R	e	P	C
SOM	87.2%	N/A	0.87	13.2%	2.39	89.9%	N/A	0.85	11.8%
SOM-SD	95.29%	0.73	0.81	95.3%	11.96	95.28%	0.73	0.80	93.9%

Comparison



Conclusions

- Graph structured data can be processed by Recursive NN – robust to noise both in label and structure; can accept numerical labels
- Higher complexity, and restricted class of graphs (trees)