Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence, Silicon Valley, USA, Nov 2-5, 2007, pp299-302.

Growing Hierarchical Self-Organizing Maps for Web Mining

Joseph P. Herbert, JingTao Yao
Department of Computer Science
University of Regina, Regina, Canada, S4S 0A2
{herbertj, jtyao}@cs.uregina.ca

***Abstract*— Many information retrieval and machine learning methods have not evolved in order to be applied to the Web. Two main problems in applying some machine learning techniques for Web mining are the dynamic and ever-changing nature of Web data and the sheer size of possible dimensions that this data could portray. One such technique, self-organizing maps (SOMs), have been enhanced to deal with these two problems individually. The growing hierarchical self-organizing map can adapt to the dynamic data present on the Web by changing its topology according to the amount of change in input size. In addition, it reduces local dimensionality by splitting features into levels. We extend this model by including bidirectional update propagation over the levels of the hierarchy. We demonstrate the effectiveness of the new approach with a Web-based news coverage example.**

I. Introduction

Knowledge discovery over the Web, or Web mining, is divided into three domains of study: Web content mining, Web usage mining, and Web structure mining [7]. These areas utilize many traditional information retrieval (IR) and data mining techniques [5], [9], [14] in the Web domain.

Self-organizing maps (SOMs) [6] are an approach to discovering similar patterns found within vector data [4]. Used to cluster attribute data for pattern recognition, the SOM model has many configurable aspects to suit different applications.

The self-organizing map is somewhat capable of performing knowledge discovery on the Web. Some SOM applications to Web content mining (deriving useful information from Web pages) include Web page and document clustering and document retrieval [2], [8]. A recommendation system using SOM clusters [15] is an application to Web usage mining (deriving information regarding a set of Web users' characteristics).

Although the above research has had some exposure, there are still problems that should be solved in order for the full potential of SOMs is realized. The first problem is the dynamic nature of Web data. In order to adequately classify data and give a low dimensional view of high dimensional data, the SOM must be trained on a finite data set that represents future data to be used. The growing self-organizing map, proposed by Villmann *et al* [13], analyzes the current collection of neurons and determines if dimensions are needed to be added or subtracted in order to improve the entire network's ability to classify data.

The second problem facing SOMs is that of high dimensionality of features. The Web is a vastly immense collection of documents. It is near impossible to have input vectors that contain all possible features. Hierarchical self-organizing maps [12] is a different approach to reducing the search space. Classification is performed on a level-by-level basis.

Seeing a need for an integrated approach, the growing hierarchical self-organizing map [1], [10] was proposed to take advantage of the benefits that the individual growing and hierarchical methods offered. However, it is unable to reflect changes to preserve the hierarchical relationships between levels. We extend this method by introducing bidirectional propagation of neuron updates over multiple hierarchy levels. This ensures that new information is correctly represented throughout the entire system. We look at how this approach is suitable for Web mining by looking at a Web-based news coverage example.

II. Web Mining Model for Self-Organizing Maps

This section introduces new extensions to the growing hierarchical self-organizing map. The extensions incorporates previous ideas of growing SOMs and integrates them with a level-wise updatable hierarchical SOM model, or bidirectional propagation of updates.

A. Feature Map Specification

There are four total layers in the model, shown visually in Figure 1. Once input has been presented to the network through the Input Layer, a suitable level in the hierarchy of SOMs is found. That is, a level whose neurons have a collectively maximum similarity to the input. This SOM is then passed on to the Growth Layer which determines whether additional neurons need to be added or existing neurons need to be removed. Once a suitable neuron has been chosen, it is passed on to the Update Layer updates the corresponding neuron and its neighbourhood, as well as level-wise updates to parents and children within the hierarchy.

A growing hierarchical SOM is formally defined as a set of hierarchy levels $A = \{A_1, \ldots, A_t\}$, where $A_i = \{W_{i,1}, \ldots, W_{i,m}\}$ is a set of SOMs. Let $W_{i,j} = \{w_1, \ldots, w_n\}$ be a SOM with n neurons. For each $w_k \in W_{i,j}$, it contains a storage unit s_k and a weight vector $\mathbf{v}_k$. Therefore, each neuron has the structure $w_k = \{\mathbf{v}_k, s_k\}$. There are also three functions

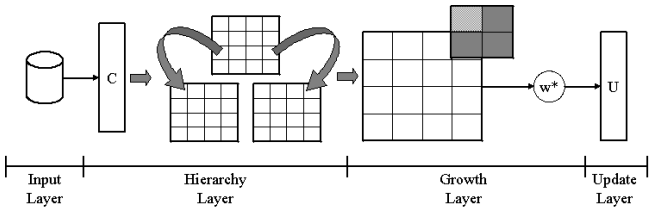

Fig. 1. The growing hierarchical self-organizing map model for Web mining.

that are introduced. The *Lev*( ) function returns the hierarchy level that a SOM currently resides on. The functions *Par*( ) and *Chd*( ) take neurons as arguments are return the parent SOM and set of children SOMs respectively.

Each hierarchy level A contains at least one SOM W. At the top-most hierarchy level, A_1 would contain exactly one SOM. This map contains the absolute highest concept view of the entire hierarchical structure. The second hierarchy level, denoted A_2, potentially contains multiple SOMs. For example, let $A_2 = \{W_{2,1}, W_{2,2}, W_{2,3}, W_{2,4}\}$ be the second level in the hierarchy with four individual SOMs. Additional SOMs on subsequently lower hierarchies are denoted by the sequence of parent SOMs, i.e. $W_{2,6,4}$ denotes that this map is the fourth such map on the third level of the hierarchy (A_2) derived from the sixth map on the previous level.

The hierarchical structure of the growing hierarchical SOM allows it to determine which feature map would best describe a relationship between a subset of inputs. Starting at the highest level in the hierarchy (root), we descend the structure to find a neuron whose weight vector is closest to the input vector. If the error rate is sufficient between the neurons and input at this level, we update the neuron and its neighborhood. We propagate the updates downwards and upwards in the hierarchy. However, if there exists a neuron a subsequent level that has a better error value, we should proceed to that level and start the process over again, finding a new neuron. We continue this process until we find the neuron whose weight vector has the lowest error between itself and the input.

The growing methods allow the topology of a single feature map to change according to new information. The error rate ε is accumulated for a particular winning neuron. If it is above an upper-bound error threshold ε_u, there is an insufficient amount of neurons representing that concept in current feature map. Therefore, a neuron is added to the outer edge of the concept cluster, with weight vectors initialized accordingly. In contrast, if the measured error rate is below a lower-bound error threshold ε_l, a neuron may be removed from the network.

The actual error rate measured from the winning neuron, ε_A, is defined as $\varepsilon_l < \varepsilon_A < \varepsilon_u$. If $\varepsilon_A < \varepsilon_l$, the SOM representation of input is an overfit. Likewise, if $\varepsilon_u < \varepsilon_A$, the SOM representation of input is an underfit.

When calculating the error, the number of "victories" a neuron has achieved must be gathered. Additional storage for the number of victories it has achieved is requried. This enables the system to calculate the popularity of a particular cluster or concept area is within a SOM.

B. Learning of Features through Bidirectional Propagation

A SOM must be trained on a subset of data before the map is considered applicable. To find the neuron $w_i \in W$ that has a weight vector closest to $\mathbf{p}_k$, similarity measures [11] are observed between each neuron and the input vector. A neuron w_i^* is marked as the winner for input vector $\mathbf{p}_k$ if it has the highest similarity value to the input vector.

Once a winning neuron has been identified, its weight vector must be updated according to the learning rate α. The value of α decays over time according to an iteration q. This ensures that the system learns features quickly at the beginning of a session and progressively moves towards precise learning as training continues. This process is done by computing the Kohonen rule [6], shown in Equation (1),

$$\mathbf{v}_i^*(q) = \mathbf{v}_i^*(q-1) + \alpha(\mathbf{p}_k(q) - \mathbf{v}_i^*(q-1)). \qquad (1)$$

The weight vector for the winning neuron w_i^* (denoted by the asterisk) at iteration q is equal to the original weight vector at iteration $(q-1)$ plus the α-scaled difference between the current input vector $\mathbf{p}_k$ and the original weight vector $\mathbf{v}_k$.

The neighbourhood must then be updated. The neighbourhood set is calculated around w_i^* according to the decaying neighbourhood distance d. A modified learning rate $\alpha\prime$ is used on the neurons within the neighbourhood set $N_{i^*}(d)$ [3], shown in Equation (2),

$$\begin{aligned}\mathbf{v}_{N_{i^*}(d)}(q) &= \mathbf{v}_{N_{i^*}(d)}(q-1) + \\ &\quad \alpha\prime(\mathbf{p}_k(q) - \mathbf{v}_{N_{i^*}(d)}(q-1)). \qquad (2)\end{aligned}$$

Each neuron $w_i \in W$ has a neighbourhood $N_i(d)$ associated with it, where each neuron's proximity is within that defined by d, a scalar value that is changed according to an iteration q. For each neuron w_i, the neighborhood $N_i(d) = \{w_r, \ldots, w_s\}$ consists of all neurons that have connectivity to w_i within distance d. An iteration q is completed when all input vectors have been introduced to the competition layer, a neuron has been selected as the winner, and the update layer has completed.

Utilizing ideas from both growing SOMs and hierarchical SOMs, we introduce extensions to the growing hierarchical SOM model suitable for Web Mining. With this model, high-dimensional data prevalent throughout the Web are able to be abstracted through a hierarchy of SOMs that offer high-level views of feature subsets. The static training model from the traditional SOM model is done away with in preference to the dynamic nature of the growing SOM.

A new definition of a neuron is used in order to provide links between hierarchies in a hierarchical SOM. For any given neuron on hierarchy A_2, there is a link to the parent hierarchy A_1 and a link to the child hierarchy A_3.

Let $w_i^* \in W_{j,k}$ be the winning neuron for input k. To propagate updates to this neuron upwards in the hierarchical structure, we calculate *Par*(w_i^*)= $W_{j-1,m}$, where *Lev*($W_{j-1,m}$) < *Lev*($W_{j,k}$). For all neurons $w_a \in W_{j-1,m}$ that are similar to w_i^*, update the corresponding weight vectors,

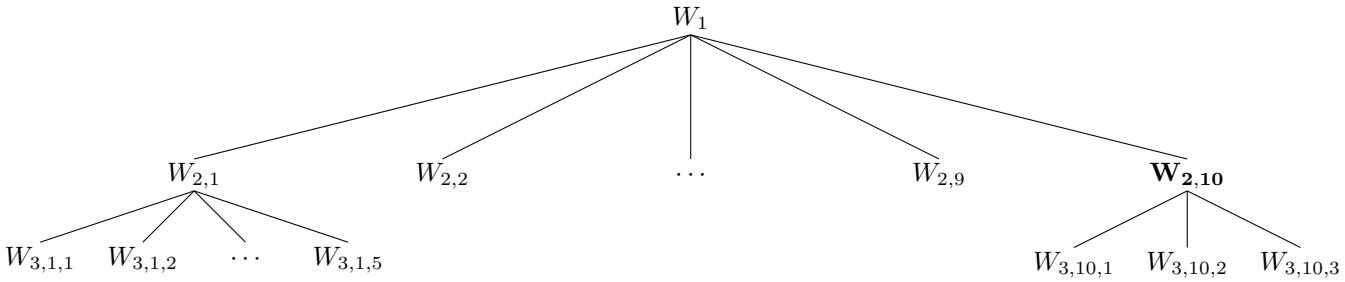

Fig. 2. The SOM hierarchy for the Online News Site.

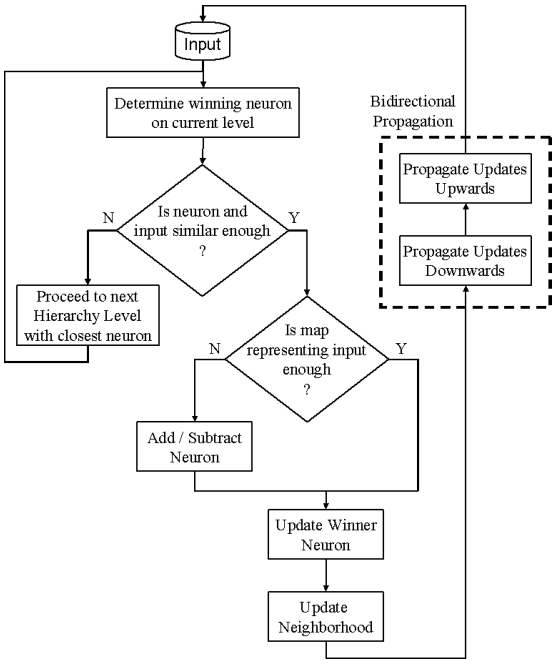

Fig. 3. The process flow of the Web mining self-organizing map model.

$$\mathbf{v}_a^*(q) = \mathbf{v}_a^*(q-1) + \beta(\mathbf{p}_k(q) - \mathbf{v}_a^*(q-1)). \quad (3)$$

To propagate updates downwards in the hierarchical structure, we calculate *Chd*(w_i^*)= A_{j+1}^*, where $A_{j+1}^* \subseteq A_{j+1}$ and $j+1$ signifies the next level in the hierarchy succeeding level j. For all neurons $w_b \in W_{j+1,t}$, where $W_{j+1,t} \in A_{j+1}^*$, update the corresponding weight vectors,

$$\mathbf{v}_b^*(q) = \mathbf{v}_b^*(q-1) + \gamma(\mathbf{p}_k(q) - \mathbf{v}_b^*(q-1)). \quad (4)$$

The learning rates β and γ are derived from a value of α. Generally, updates to a parent neuron on hierarchy level $j-1$ are not as strong as updates to children neurons $j+1$ for a given neuron on level j. The relationship $\beta < \alpha < \gamma$ allows for the propagation upwards and downwards to reflect the proper weight vector magnitude change. The process flow of the entire SOM model, which includes the sequential operation of the hierarchical and growing operations, is shown in Figure 3.

III. Web-based News Coverage Example

An example of how the extended growing hierarchical SOM model can be used for Web content mining is shown here. An online news site could make use of the hierarchical model to organize news articles in a logical way. A news site has the potential to push to the user thousands of articles pertaining to many areas, such as global news, politics, technology, etc.

In order to use a SOM for this application, each news document is preprocessed into a vector format that can be presented to the feature map. Each component in the input vector is an indication of how frequent a particular word occurs in a news article. The term frequency (*tf*) measure is useful for this purpose. These input vectors contain information regarding the frequency of each word within the document.

For example, an input vector $\mathbf{p}_k = \{c_1, c_2, c_3\}$ has three components (keywords). Let us say the keywords {*pharmaceutical*, *insurance*, *disease*} are used to describe a Health-related news document. Therefore, c_2 is a measure of how often the keyword *insurance* appears in document k. In general, component c_i is a *tf*-measure of the ith keyword in news document k.

At the top-most level in the hierarchy, news articles pertaining to high-level concepts are organized according to their features. The entire collection of documents on the online news site are presented through feature maps that abstract their similarities. They are organized as the following equations and presented as a hierarchy in Figure 2.

$$\begin{aligned}
A_1 &= \{W_1\}, \\
A_2 &= \{\{W_{2,1}\}, \{W_{2,2}\}, \{W_{2,3}\}, \{W_{2,4}\}, \{W_{2,5}\}, \\
&\quad \{W_{2,6}\}, \{W_{2,7}\}, \{W_{2,8}\}, \{W_{2,9}\}, \{W_{2,10}\}\}, \\
A_3 &= \{\{W_{3,1,1}, W_{3,1,2}, W_{3,1,3}, W_{3,1,4}, W_{3,1,5}\}, \\
&\quad \{W_{3,2,1}, W_{3,2,2}\}, \\
&\quad \{W_{3,3,1}, W_{3,3,2}, W_{3,3,3}, W_{3,3,4}, W_{3,3,5}\}, \\
&\quad \vdots \\
&\quad \{W_{3,10,1}, W_{3,10,2}, W_{3,10,3}\}\}, \quad (5)
\end{aligned}$$

where W_1 is the highest level map of the hierarchy, or *Lev*(W_1)= 1 . The individual maps $W_{2,1}, \cdots, W_{2,10}$ on the second hierarchy level A_2 are Web documents pertaining to Global news, Local news, Politics, Business, Weather, Entertainment, Technology, Sports, Opinions, Health respectively.

These maps can be derived by taking a neuron $w_i \in W_1$ and executing *Chd*(w_i). Descending through the hierarchies to the third level A_3 reveals more SOMs. These are derived from the previous higher level SOMs. The SOM set $\{W_{3,10,1}, W_{3,10,2}, W_{3,10,3}\}$ contain individual SOMs relating to specific sub-areas within *Health*, the parent concept represented in $W_{2,10}$.

To illustrate the results of the *Par*() and *Chd*() functions, let us look at a the hierarchy presented in Figure 2. For a neuron $w_i \in W_{2,1}$, *Par*(w_i)= W_1 results in *Lev*(W_1) < *Lev*($W_{2,1}$). Likewise, for a neuron $w_i \in W_{2,1}$, we can execute *Chd*(w_i)= $\{W_{3,1,1}, \ldots, W_{3,1,5}\}$. If we let $A_k = \{W_{3,1,1}, \ldots, W_{3,1,5}\}$, we have $A_k \subseteq A_3$, where *Lev*($W_{2,1}$) < *Lev*($W_{3,1,i}$).

The SOM $W_{2,10}$ is presented in Figure 4. Clusters have been labeled to show relationships to lower levels. The maps $W_{3,10,1}$, $W_{3,10,2}$, and $W_{3,10,3}$ are derived from their common parent map $W_{2,10}$. One of the key advantages of this hierarchical structure is that as we descend through the levels, more neurons are used to decrease the level of abstraction on a particular topic. The map shown expands on the previous 8 neurons and organizes documents with 90 neurons. We can see that more distinct similarities between documents can be expressed as we increase the number of neurons. That is, perhaps a subset of neurons in $W_{2,10}$ relate to *Health Research Funding*, further expanded in map $W_{3,10,1}$. Additionally, the 26 neurons pointing to hierarchy map $W_{3,10,2}$ pertain to *Health Outbreak Crises* articles.

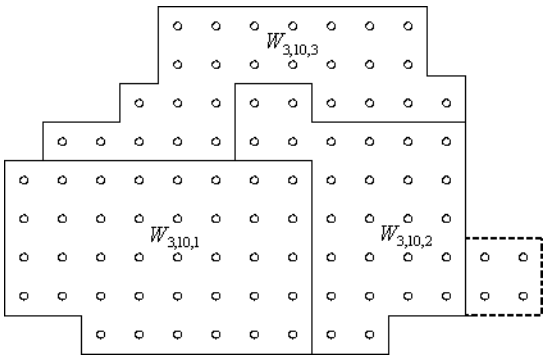

Fig. 4. The feature map $W_{2,10}$ of hierarchy A_2 shown in Figure 2. Clusters with neurons that link to the next lowest level of the hierarchy are labeled.

When adding new articles, the input vector representation of the article would be compared first the top-most feature map using the *tf*-measures in the components. Descending through the tree until a similarity measure has been maximized results in finding the correct hierarchy to represent this document. The neurons in the feature map would be updated to reflect the addition of this article. For example, a new *Health*-related article describing a recent outbreak in a country has been received. Descending through our hierarchy, we find that it should have representation in the SOM $W_{3,10,2}$. The addition of neurons in $W_{3,10,2}$ to better reflect this crisis is performed. This is shown by the addition of four neurons in the dashed area in Figure 4.

IV. Conclusion

The extended growing hierarchical self-organizing map model for Web mining introduced in this paper incorporates the previous approaches that can help in minimizing the impact of a training procedure and allowing for different levels of abstraction to reduce feature vector dimensionality.

The Web-based news coverage example demonstrates the strengths of this approach. The hierarchical structure of SOMs can be used to classify Web documents with natural language by reducing dimensionality. In addition, the dynamic nature of Web data can cause the SOM to change its topology.

The bidirectional propagation within this new approach allows for new information learned by an individual SOM to be reflected in other hierarchy levels. Update propagation upwards reflect how a parent SOM partially changes in view of a change in a child SOM. Propagation downwards reflect how children SOMs are more fully influenced by changes to a parent SOM. The extension of the growing hierarchical self-organizing map shown in this article is useful for the next-generation of Web-enabled systems.

References

[1] M. Dittenbach, A. Rauber, and D. Merkl, "Uncovering hierarchical structure in data using the growing hierarchical self-organizing map," *Neurocomputing*, vol. 48, pp. 199–216, 2002.

[2] A. Georgakis, C. Kotropoulos, A. Xafopoulos, and I. Pitas, "Marginal median som for document organization and retrieval," *Neural Networks*, vol. 17, pp. 365–377, 2004.

[3] J. Herbert and J. T. Yao, "A game-theoretic approach to competitive learning in self-organizing maps," in *International Conference on Natural Computation*, vol. 1, 2005, pp. 129–138, LNCS 3610.

[4] T. L. Huntsberger and P. Ajjimarangsee, "Parallel self-organizing feature maps for unsupervised pattern recognition," *International Journal of General Systems*, vol. 16, no. 4, pp. 357–372, 1990.

[5] V. Kecman, *Learning and Soft Computing, Support Vector Machines, Neural Networks and Fuzzy Logic Models*. The MIT Press, 2001.

[6] T. Kohonen, "Automatic formation of topological maps of patterns in a self-organizing system," in *Proceedings of the Scandinavian Conference on Image Analysis*, 1981, pp. 214–220.

[7] R. Kosala and H. Blockeel, "Web mining research: A survey," *SIGKDD Explorations*, vol. 2, no. 1, pp. 1–15, 2000.

[8] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen, "Websom for textual data mining," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 345–364, 1999.

[9] L. Ramirez, N. G. Durdle, V. J. Raso, and D. L. Hill, "A support vector machines classifier to assess the severity of idiopathic scoliosis from surface topography," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 84–91, 2006.

[10] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1331–1341, 2002.

[11] S. Santini and R. Jain, "Similarity measures," *IEEE Transactions: Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871–883, 1999.

[12] P. N. Suganthan, "Hierarchical overlapped som's for pattern classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 1, pp. 193–196, 1999.

[13] T. Villmann and H. Bauer, "Applications of the growing self-organizing map," *Neurocomputing*, vol. 21, pp. 91–100, 1998.

[14] L. P. Wang and X. J. Fu, *Data Mining with Computational Intelligence*. Springer, 2005.

[15] X. Z. Wang, A. Abraham, and K. A. Smith, "Intelligent web traffic mining and analysis," *Journal Of Network And Computer Applications*, vol. 28, no. 2, pp. 147–165, 2005.