Online incremental hierarchical classification resonance network

Ju-Youn Park, Jong-Hwan Kim

A School of Engineering and Applied Science, George Washington University, Washington, DC 20052, United States
B School of Electrical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Daejeon 34141, Republic of Korea

A R T I C L E   I N F O

Article history:
Received 1 November 2019
Revised 25 June 2020
Accepted 22 September 2020

Keywords:
Hierarchical classification
Incremental class learning
Blocking method
Human-Computer/robot interaction
Multimedia recommendation

A B S T R A C T

Hierarchical classification is imperative in that almost all objects are described in hierarchical semantics. If a classification method enables incremental class learning to learn new objects online, it will be practically used for real-time applications. In this sense, we propose online incremental hierarchical classification resonance network (OIHCRN) that enables online incremental class learning in hierarchical classification. OIHCRN has a structure that grows horizontally and vertically online according to object classes, so that a newly added object can be classified. By the proposed process of scale-preserving projection and prior label appending, OIHCRN reflects the class dependency between class levels and simultaneously normalizes the input vector online. Additionally, to reduce the model complexity and improve performance, two auxiliary strategies, named OIHCRN with class END and OIHCRN with differentiated class labels, are introduced. To demonstrate the effectiveness of OIHCRN, experiments are carried out for benchmark datasets and then for a multimedia recommendation system.

1. Introduction

Classification is one of the most important problems in machine learning. For real time applications, classification should be able to learn new knowledge incrementally and reflect it in response to the changing environment. In other words, incremental class learning is required for classification so that a new class can be classified without re-training for all the data [1–4]. Since objects can be described in a hierarchical semantics, it is imperative to enable incremental class learning in hierarchical classification which classifies a given data belonging to classes that are organized into a class hierarchy [5]. However, existing hierarchical classification methods do not allow incremental class learning because they require a predefined class hierarchy and/or classes that make up the hierarchy. For example, deep learning-based methods, which have achieved promising results, mainly include the last layer composed of output nodes as many as the number of object classes in the dataset used for training, and thus cannot classify when a new object class is given. Even if these methods increase the number of nodes, performance decreases because they are influenced by the previously learned knowledge, that is, the weight values. In this sense, an adaptive resonance theory-supervised predictive mapping for hierarchical classification (ARTMAP-HC) was devised as the first approach to enable incremental class learning in hierarchical classification [6].

Since ARTMAP-HC is based on fuzzy ARTMAP [7], however, ARTMAP-HC undergoes the node proliferation issue that is occurred in fuzzy ARTMAP. Furthermore, ARTMAP-HC additionally introduces the online normalization process in advance to handle an input vector online without prior knowledge of all data because the attribute values of an input value should be normalized into [0,1] for fuzzy ARTMAP. This paper proposes a new network, online incremental hierarchical classification resonance network (OIHCRN), to enable online incremental class learning in hierarchical classification with high performance. The proposed network employs a novel classification network, OICRN, which enables incremental class learning online [8], for hierarchical classification. OICRN mitigates the node proliferation issue in fuzzy ARTMAP. Accordingly, OIHCRN also alleviates the problem. OICRN is able to normalize an input vector online by the scale-preserving projection process, and thus OIHCRN does not need to require an additional process beforehand. Besides, by using OICRN which showed superb performance with the incremental class learning ability, OIHCRN show improved performance.

OIHCRN is composed of hierarchically stacked modules like ARTMAP-HC, and each module incorporates two OICRNs. One is OICRN for blocking, which determines whether to perform an additional classification for a higher level class, and the other is OICRN for classification, which classifies the corresponding class for a certain level depending on the given input. OIHCRN enables
incremental class learning because hierarchically stacked modules as well as the number of class nodes grow online, and there is no reduction in performance because they preserve the already learned knowledge. OIHCRN employs a process of scale-preserving projection and prior label appending which simultaneously normalizes an input online and manifests the dependency between class levels [9–11].

As mentioned earlier, each module in OIHCRN, consisting of two OICRNs, is associated with a class level, and the weights of the two OICRNs can be correlated. Therefore, to reduce the model complexity, two additional methods to merge the two OICRNs are introduced into the original OIHCRN. By introducing the variants, two separate OICRNs are merged and the correlation that exists between the networks implicitly affects the blocking process, which is expected to improve blocking and improve classification performance. OIHCRN with class END (OIHCRN-CE) introduces class END indicating that all the hierarchical classes are labeled so that classification is over at the previous class level. By building a module for an additional level and labeling the class END at the added level, a given input can be classified into a class hierarchy without two separate OICRNs. OIHCRN with differentiated class labels (OIHCRN-DL) introduces differentiated class labels and allows to classify a given input into a class hierarchy without building an additional level module. The differentiated class labels divide each class into two labels, depending on whether the class is labeled at the end of the class level.

To the best of our knowledge, ARTMAP-HC has been the first and only approach to enable incremental class learning in hierarchical classification, and OIHCRNs outperform ARTMAP-HC. In addition to ARTMAP-HC, other hierarchical classification methods that do not allow incremental instructional learning but are expected to perform better with offline training are used to compare with our networks. Compared with other methods where incremental class learning is not possible, we can prove that OIHCRNs can be practically used if the proposed networks after training for each new class provided sequentially show similar or even better performance. Among OIHCRNs, OIHCRN-DL shows the best performance. For OIHCRN-DL, which has the highest performance, the incremental class learning ability is also validated by training the network sequentially for each instance of the training data and testing on all the test data instances each time a data instance is trained. Furthermore, the distributions of the highest levels of targeted and classified test data instances respectively by OIHCRN, OIHCRN-CE, and OIHCRN-DL are measured across the cross-validation process and compared to analyze why the performances were changed between the networks.

OIHCRN is also applied to a multimedia recommendation system for digital storytelling [6,12]. A digital companion, a software agent that resides in a smartphone and provides emotional interactive services to a user. If the digital companion can communicate with the user using additional multimedia as well as sentences generated from the conversation, the user can understand what the digital companion has intended more effectively. OIHCRN classifies the appropriate media by its type and then by the specific genre hierarchically based on the keywords of a dialogue and user context information, and the corresponding media is provided to the user while talking with a digital companion. Demonstrations of the interactions between a user and a digital companion are performed, and how the multimedia recommendation system is applied is shown.

The rest of this paper is organized as follows. Section 2 briefly describes OICRN which is the basic network of the proposed OIHCRN. Section 3 proposes OIHCRN, OIHCRN-CE, and OIHCRN-DL. Section 4 presents the experimental results, and Section 5 describes a multimedia recommendation system for digital storytelling. Finally, concluding remarks follow in Section 6.

2. Preliminary

2.1. OICRN

OICRN enables incremental class learning in multi-class classification online with high performance [8]. OICRN incorporates the input field $P^i$, the cluster field $P^C$, and the map field $P^M$, as shown in Fig. 1. Assume an input vector $a = (a_1, a_2, \cdots, a_n)^T$, and the corresponding class K are given. The proposed scale-preserving projection process allows classifying raw input vectors online without a normalization process in advance as follows:

$$a^+ = (a^T 1)^T = (a_1, a_2, \cdots, a_n, 1)^T$$

$$I = \frac{a^+}{||a||}$$  \hspace{1cm} (1)

where $a$, $a^+$, and $I$ denote an input vector, the dimension-expanded vector, and the projected input vector, respectively.

In OICRN, the concept vector, defined as the unit normalized mean vector of the vectors in a cluster [13], is employed as a representative vector for each associated cluster. Through the scale-preserving process, the input vectors exist on the unit sphere, $R^d$ where $d = N_0 + 1$, and they are clustered into $k$ disjoint cluster nodes, $\pi_1, \pi_2, \cdots, \pi_k$. For a cluster node $\pi_j$, the mean vector or centroid of the vectors in the cluster node is obtained as follows:

$$m_j = \frac{1}{n_j} \sum_{k=\pi_j} I$$  \hspace{1cm} (2)

where $n_j$ is the number of vectors in $\pi_j$. The concept vector is defined as the unit normalized vector of the mean vector as follows:

$$c_j = \frac{m_j}{||m_j||}$$  \hspace{1cm} (3)

where $|| \cdot ||$ denotes the L2-norm.

The cosine similarity between the input vector and each concept vector $c_j$ is calculated for clustering using the following choice
function:

\[ T_j(I) = \Gamma^T c_j. \quad (4) \]

After the cosine similarities between the input vector and existing concept vectors are calculated, the closest cluster node \( \pi_j \) that obtains the largest value from the choice function is selected in the cluster field \( F \) as follows:

\[ J = \arg \max_j \{ T_j(I) : j = 1, \ldots, N_\pi \} \quad (5) \]

where \( N_\pi \) is the number of cluster nodes. The cluster node \( \pi_j \) is finally selected only if the corresponding concept vector \( c_j \) is reasonably close to \( I \), in other words, only if the match function \( M_j(I) \) representing the resonance value is larger than the vigilance parameter \( \rho \) as follows:

\[ M_j(I) = \frac{n_j \alpha + n_j^I c_j}{\alpha + n_j^I} \geq \rho \quad (6) \]

where \( \alpha \) is the regularization parameter that regularizes the effect of \( n_j \) on \( M_j(I) \).

In the map field \( F_M \), the weight vector \( w_j^M \) associated with the selected cluster node \( \pi_j \) is compared to the given label vector \( b = (b_1, b_2, \ldots, b_{n_y})^T \) where \( b_k = \begin{cases} 1 & \text{if } k = K \\ 0 & \text{if } k \neq K \end{cases} \). If \( w_j^M \) and \( b \) are not the same, the value of vigilance parameter \( \rho \) is increased by \( \Delta \rho \) until the following condition is satisfied for \( I \) to search for another cluster node:

\[ \rho > \frac{n_j \alpha + n_j^I c_j}{\alpha + n_j^I} \quad (7) \]

where the cluster node \( \pi_j \) is activated in the cluster field. If there is no cluster node that satisfies the resonance condition, then a new cluster node is generated with \( w_j^M = b \). Through the resonance process, the input vector which belongs to a new class can be classified. In \( F_M \), the weight vector \( w_j^M \) associated with the selected cluster node \( \pi_j \) is updated with the label vector \( b \).

3. OICRNC

OICRNC enables online incremental class learning in hierarchical classification. The hierarchical label vector \( b = (b_1, b_2, \ldots, b_{n_y})^T \) is specified to be classified for the given input vector \( a = (a_1, a_2, \ldots, a_{n_y})^T \). \( b \) is composed of associated hierarchical labels where each element \( b_i \) of \( b \) represents the index of a label \( K_i^c \) for the corresponding class level \( l \), and \( M_b \) is the highest class level. As Fig. 2 shows, OICRNC is composed of hierarchically stacked modules, and each module incorporates two OICRNs, OICRN_{\phi} for blocking and OICRN_{\pi} for classification.

The blocking label \( K_b^\phi \) for the corresponding level \( l \) is set as

\[ K_b^\phi = \begin{cases} 0 & \text{if } l < M_b \\ 1 & \text{if } l = M_b \end{cases} \]

3.1. Scale-preserving projection and prior label appending

Since OICRN exploits concept vectors that are unit vectors and we normalize an input vector by the scale-preserving projection process (1), raw data can be classified online. However, even
though class level increases, the norm value of the input vector should be 1 and the input vector should also include the information of prior label. To do so, we propose the process of scale-preserving projection and prior label appending. The proposed process utilizes a rotation matrix to insert the class level dependency information into the input vector and normalize the resulted vector as a unit vector at the same time.

Generally, rotations are described as rotations around an axis. However, a more correct way to conceptualize rotations is to consider them as rotations parallel to a given plane and this view is valid in 2 dimensions, 3 dimensions, and even in any higher dimensions [14]. General rotations in N dimensions can be viewed as a sequence of elementary rotations and each elementary rotation acts in the plane of a particular pair, (i, j), of coordinates, leaving an (N − 2)-dimensional subspace unchanged [15]. The rotation parallel to the (N − 1, N)-plane can be represented as a rotation matrix as follows:

\[
\mathbf{R}_{(N-1,N)} = \begin{bmatrix}
1 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 1 & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & \cos \theta & -\sin \theta \\
0 & 0 & 0 & \ldots & 0 & \sin \theta & \cos \theta
\end{bmatrix},
\]

The process of scale-preserving projection and prior label appending employs an elementary rotation. In order to preserve the scale component even after the normalization, we extend one dimension to the input vector \(\mathbf{a}\) and set 1 to the added element. We extend additional dimension with the added element 0 because an elementary rotation is performed parallel to the plane formed by two elements. After the modified scale-preserving projection with 2-dimension expansion, we project the expanded vector \(\mathbf{a}^+\) onto the unit sphere so the projected vector \(\mathbf{I}\) has unit norm as follows:

\[
\mathbf{a}^+ = (\mathbf{a}^T \ 1 \ 0)^\top,
\]

\[
\mathbf{I} = \frac{\mathbf{a}^+}{||\mathbf{a}^+||}.
\]

The projected vector \(\mathbf{I}\) is fed into OIHCRN as an input vector. The appended input vector of OIHCRN, \(\mathbf{I}'\) consists of the scale-preserving projected vector \(\mathbf{I}\) through the modified projection process (9) and the label that is classified at the prior level \(l-1\). Specifically, rotating the projected vector \(\mathbf{I}\) parallel to the \((N_0 + 1, N_0 + 2)\)-plane by \(\theta_{b_{l-1}} = \frac{\pi}{b_{l-1}}\) forms the prior label appended vector as follows:

\[
\mathbf{I}' = \mathbf{R}_{b_{l-1}}^{N_0+1} \mathbf{I} = \begin{bmatrix}
1 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 1 & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & \cos \theta_{b_{l-1}} & -\sin \theta_{b_{l-1}} \\
0 & 0 & 0 & \ldots & 0 & \sin \theta_{b_{l-1}} & \cos \theta_{b_{l-1}}
\end{bmatrix} \begin{bmatrix}
a_1/z \\
a_2/z \\
a_3/z \\
\vdots \\
a_{N_0}/z \\
\end{bmatrix} = \frac{1}{z} \begin{bmatrix}
(a_1, a_2, a_3, \ldots, a_{N_0}, \cos \theta_{b_{l-1}}, \sin \theta_{b_{l-1}}) \\
\end{bmatrix}^\top.
\]

\[
z = \sqrt{a_1^2 + a_2^2 + \cdots + a_{N_0}^2 + 1}.
\]

With the label information of prior level contained in a rotating angle of \(\theta_{b_{l-1}} = \frac{\pi}{b_{l-1}}\), OIHCRN can reflect the dependencies between the prior labels and the corresponding class label in the associated class level. By including the predicted class label from the prior level as the denominator of the rotation angle, the classification module for the current level can predict the associated class label taking into account the class label of the previous level. In that sense, OIHCRN can reflect label dependencies between consecutive class levels. Since two dimensions expanded beforehand, the scale-component of the original input vector is preserved and the prior label information can be included without distorting features in the original input vector. The appended vector is the unit vector because a rotation does not change the scale of the vector. Thus, a subsequent process for normalization is not needed. Furthermore, in order to perform the proposed process of scale-preserving projection and prior label appending, only two triangular functions are needed to be calculated. An example of the process for given one-dimensional input vectors is depicted in Fig. 3.

3.2. Learning procedure

As described in Algorithm 1, the basic learning process for each level \(l\) is iteratively performed if the blocking label of level \(l-1\), \(K_{b}^{l-1}\) is equal to 0. In other words, if \(K_{b}^{l-1} = 0\) is satisfied, OICRN\(_{b}\) for blocking and OICRN\(_{l}\) for classification are newly built for level \(l\) to constitute OIHCRN\(_{b}\) and trained for the given input. Specifically, OICRN\(_{b}\) is activated for the appended input vector \(\mathbf{I}'\) with \(K_{b}^l = 0\) if there exists a higher level class. Otherwise, OICRN\(_{b}\) is activated with \(K_{b}^l = 1\). OICRN\(_{b}\) is then activated with \(K_{b}^l = b_l\) where \(b_l\) is the index of a label for the corresponding class level \(l\). Following the learning rule of OICRN [8], OICRN\(_{b}\) and OICRN\(_{l}\) update the weight vectors.
Algorithm 1 OIHCRN Learning Procedure.

1: Let $z = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} + 1$
2: Scale-preserving Projected Input Vector:

\[ I = \frac{\mathbf{b}}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}} \]
3: Hierarchical Label Vector: $\mathbf{b} = (b_1, b_2, \ldots, b_M)^T$
4: $I = 1$
5: repeat
6: if $I = 1$ then
7: $I = 1$
8: else
9: $\mathbf{r} = (I - 1) \mathbf{b} \cdot \mathbf{b}^T$ where $\mathbf{r} = 2a_1$
10: end if
11: if $I = M_b$ then
12: Activate OICRN$_b$ according to $\mathbf{r}$ and $K_0 = 0$
13: else
14: Activate OICRN$_c$ according to $\mathbf{r}$ and $K_0 = 1$
15: end if
16: Perform the match tracking for OICRN$_b$
17: Update the weight vector of OICRN$_c$
18: Activate OICRN$_c$ according to $\mathbf{r}$ and $K_0 = b_1$
19: Perform the match tracking for OICRN$_c$
20: Update the weight vector of OICRN$_c$
21: $I = I + 1$
22: until $K_0 = 0$

3.3. OIHCRN-CE

As described in the previous section, the proposed OIHCRN is composed of a base module with two OICRNs, OICRN$_b$ and OICRN$_c$ at each class level. Combining the separate OICRNs can reduce the model complexity. Therefore, two methods are introduced to build one OICRN per class level for hierarchical classification, and the established networks are called OIHCRN-CE and OIHCRN-DL, respectively.

The first method is to add a new class, which is called class END, indicating that all the hierarchical classes are labeled so that classification is over at the previous class level. By building a module for an additional level and labeling the class END at the added class level, a given input can be classified into a class hierarchy without two separate OICRNs. The class END is labeled as the class label $K = 1$, and the class label for each class level is increased by 1 for training. In other words, the class label for the corresponding class level $K'$ is defined as follows:

\[ K' = \begin{cases} b_l + 1 & \text{if } l \leq M_b \\ 1 & \text{if } l = M_b + 1 \end{cases} \]

where $b_l$ represents the index of the label for the corresponding class level $l$, and $M_b$ is the highest level of the given hierarchical label vector $\mathbf{b} = (b_1, b_2, \ldots, b_M)^T$. As shown in Fig. 4, when a scale-preserving projected input vector $I$ through (9) is given with a hierarchical label vector $\mathbf{b} = (b_1, b_2, b_3)^T = (3, 1, 2)^T$, OIHCRN-CE builds a four-level structure consisting of OICRN$^1$ to OICRN$^4$ and is trained using the class labels $K^1 = 4$, $K^2 = 2$, $K^3 = 3$, and $K^4 = 1$. The last class label $K^4 = 1$ indicates that classification is completed at the previous class level 3.

3.4. OIHCRN-DL

OIHCRN-CE, however, requires building an additional class level for each input, which increases model complexity. Thus, OIHCRN-DL introduces differentiated class labels and allows classifying a given input into a class hierarchy without building an additional level. The differentiated class labels divide each class into two labels, depending on whether the class is labeled at the end of the class level. OIHCRN-DL does not build an additional class level and does not create a redundant class that is not one of the actual classes, so classification performance is improved.

\[ K^l = \begin{cases} 2b_l - 1 & \text{if } l \leq M_b \\ 2b_l & \text{if } l = M_b \end{cases} \]  

(12)

where $b_l$ represents the index of the label for the corresponding class level $l$, and $M_b$ is the highest level of the given hierarchical label vector $\mathbf{b} = (b_1, b_2, \ldots, b_M)^T$. If the corresponding level is the highest level and no more level will be built, then the differentiated class label is formed into the double of the original class label index as $K^l = 2b_l$ for the corresponding level $l$. Otherwise, the differentiated class label is defined as $K^l = 2b_l - 1$ for the corresponding class level $l$. That is, each original class label is differentiated into an even or odd numbered class label, depending on whether the associated class level is highest or not.

As shown in Fig. 4, when a scale-preserving projected input vector $I$ through (9) is given with a hierarchical label vector $\mathbf{b} = (b_1, b_2, b_3)^T = (3, 1, 2)^T$, OIHCRN-CE builds a three-level structure consisting of OICRN$^1$ to OICRN$^3$ and is trained using the differentiated class labels $K^1 = 5$, $K^2 = 1$, and $K^3 = 4$. The last class label, even numbered $K^3 = 4$, indicates that classification is completed for the corresponding class level 3. The class labels at the class levels 1 and 2 are odd numbers since the given input should be classified at the additional class level.

4. Experiments

4.1. Experimental setup

In order to validate the effectiveness of the proposed OIHCRN, OIHCRN-CE, and OIHCRN-DL, we performed experiments on the
Table 1
Characteristics of the hierarchical classification datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Number of data (Train/Test)</th>
<th>Number of attributes</th>
<th>Number of classes*</th>
<th>Number of classes per level</th>
<th>Cardinality**</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIPO-alpha</td>
<td>1710 (1,352/358)</td>
<td>74,435</td>
<td>187</td>
<td>7/20/160</td>
<td>3</td>
</tr>
<tr>
<td>ImCLEF07A</td>
<td>11,006 (10,000/1,006)</td>
<td>80</td>
<td>96</td>
<td>8/25/63</td>
<td>3</td>
</tr>
<tr>
<td>ImCLEF07D</td>
<td>11,006 (10,000/1,006)</td>
<td>80</td>
<td>46</td>
<td>4/16/26</td>
<td>3</td>
</tr>
<tr>
<td>GPCR Interpro</td>
<td>7444</td>
<td>450</td>
<td>198</td>
<td>12/54/82/50</td>
<td>2.82</td>
</tr>
<tr>
<td>GPCR Pfam</td>
<td>7053</td>
<td>75</td>
<td>192</td>
<td>12/52/79/49</td>
<td>2.84</td>
</tr>
<tr>
<td>GPCR Prints</td>
<td>5404</td>
<td>283</td>
<td>179</td>
<td>8/46/76/49</td>
<td>3.01</td>
</tr>
<tr>
<td>GPCR Prosite</td>
<td>6246</td>
<td>129</td>
<td>187</td>
<td>9/50/79/49</td>
<td>2.95</td>
</tr>
<tr>
<td>EC Interpro</td>
<td>14,027</td>
<td>1216</td>
<td>330</td>
<td>6/41/96/187</td>
<td>3.66</td>
</tr>
<tr>
<td>EC Pfam</td>
<td>13,987</td>
<td>708</td>
<td>333</td>
<td>6/41/96/190</td>
<td>3.67</td>
</tr>
<tr>
<td>EC Prints</td>
<td>14,025</td>
<td>382</td>
<td>351</td>
<td>6/45/92/208</td>
<td>3.70</td>
</tr>
<tr>
<td>EC Prosite</td>
<td>14,041</td>
<td>585</td>
<td>324</td>
<td>6/42/89/187</td>
<td>3.69</td>
</tr>
</tbody>
</table>

\* The root node was not included.

** The averaged number of class labels assigned per data instance.

Fig. 5. Structure of the proposed OIHCRN-DL. $\kappa'$ denotes the differentiated class label for the corresponding class level $l$.

section D in the WIPO-alpha\(^1\) dataset [16] for text categorization, the ImCLEF07A and ImCLEF07D datasets [17] for image annotation, and 8 datasets of protein function prediction; 4 of them are formed by G-Protein-Coupled Receptors (GPCRs) and the others are formed by Enzyme Commission Codes (ECs) [18], as listed in Table 1. We used the pre-processed versions of the protein function prediction datasets in [19]. We followed the experimental setup in [6] and set $\beta$ as 0.9 for ARTMAP-HC and 0.0 for OIHCRN, OIHCRN-CE, and OIHCRN-DL, respectively. Since all the datasets have their input attribute values in [0,1], the additional normalization process was not applied for ARTMAP-HC in the experiments.

4.1.1. Comparison

Comparison experiments were conducted to verify the practical applicability by comparison with other classification methods that are not capable of incremental learning and to demonstrate the performance improvement of OIHCRN by comparison with ARTMAP-HC. We conducted experiments on BR-SVM1, BR-SVM2 [20–22], HM3 [16], Clus-HMC-Ens [23], and the methods based on the naive Bayes algorithm, local-model naive Bayes with usefulness (LMNBwU), global-model naive Bayes (GMNB), and global-model naive Bayes with usefulness (GMNBwU) [19,24]. LMNBwU uses a naive Bayesian classifier for each parent node adopting the LPBN approach. GMNB and GMNBwU adopt the global approach and use a modified Bayesian classifier for hierarchical classification to handle the class hierarchy. These methods need to be trained in advance on the entire dataset with a predefined class hierarchy to build a probability model, and incremental class learning is not possible. Therefore, before testing, the methods should be trained offline for the dataset. The methods are expected to outperform our methods because they know all the information of the classes that make up the dataset in advance. To compare with these methods, our networks were trained online for all the dataset before testing. We employed a $5 \times 2$-fold cross-validation process for all the datasets. The 2-fold cross-validation process was repeated for 5 times as in [19].

4.1.2. Incremental class learning ability

In order to verify that the proposed networks actually perform incremental class learning that incrementally learn a newly provided class without forgetting the already known classes, we conducted experiments under another setting for OIHCRN-DL which showed the highest performance among OIHCRN, OIHCRN-CE, and OIHCRN-DL without loss of generality. Specifically, it was observed how the classification accuracies of OIHCRN-DL were changed as a new training data instance was sequentially added for training. We conducted the experiments to verify the incremental class learning ability on the 4 datasets of protein function prediction for GPCRs. In addition, we observed the change in performance of OIHCRN-DL compared with ARTMAP-HC.

4.1.3. Performance analysis

To analyze why the performances were changed between OIHCRN, OIHCRN-CE, and OIHCRN-DL, the highest levels of targeted or classified test data instances were calculated, and their distributions measured across the $5 \times 2$-fold cross-validation process were compared. If the distribution for the classified test results is more similar to the distribution for the test dataset, it can be said that the blocking has been performed more appropriately. Improved blocking improves accuracy in terms of the recall. We conducted the experiments for performance analysis on the 8 datasets

of protein function prediction. To compare the model complexity, the numbers of parameters of OIHCRN, OIHCRN-CE, and OIHCRN-DL were calculated. The practical usability was also verified by measuring the training time of OIHCRN-DL for each training data instance on the 4 GPCR datasets.

4.2. Evaluation measures

The micro-averaged hierarchical precision ($P_{\mu}$), micro-averaged hierarchical recall ($R_{\mu}$), and micro-averaged hierarchical $F$-measure ($F_{\mu}$) were employed [25]. The precision ($P_i$), recall ($R_i$), and $F$-measure ($F_i$) for the class $i$ are respectively defined as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i},$$

$$R_i = \frac{TP_i}{TP_i + FN_i},$$

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i}$$

where $TP_i$ denotes the true positives, $FP_i$ denotes the false positives, and $FN_i$ denotes the false negatives. The precision, recall, and $F$-measure can be averaged in two ways: micro-averaging and macro-averaging. The micro-averaged precision ($P_{\mu}$), micro-averaged recall ($R_{\mu}$), and micro-averaged $F$-measure ($F_{\mu}$) are respectively calculated as follows:

$$P_{\mu} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)},$$

$$R_{\mu} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)},$$

$$F_{\mu} = \frac{2 \times P_{\mu} \times R_{\mu}}{P_{\mu} + R_{\mu}}$$

where the number of classes is $n$. The macro-averaged precision ($P_{M}$), macro-averaged recall ($R_{M}$), and macro-averaged $F$-measure ($F_{M}$) are respectively calculated as follows:

$$P_{M} = \frac{\sum_{i=1}^{n} P_i}{n},$$

$$R_{M} = \frac{\sum_{i=1}^{n} R_i}{n},$$

$$F_{M} = \frac{2 \times P_{M} \times R_{M}}{P_{M} + R_{M}}$$

The hierarchical classification versions of $P$, $R$, and $F$ are the hierarchical precision ($hP$), hierarchical recall ($hR$), and hierarchical $F$-measure ($hF$). $hP_j$, $hR_j$, and $hF_j$ for each data instance $j$ are respectively defined as follows:

$$hP_j = \frac{|\hat{P}_j \cap T_j|}{|\hat{P}_j|},$$

$$hR_j = \frac{|\hat{P}_j \cap T_j|}{|T_j|},$$

$$hF_j = \frac{2 \times hP_j \times hR_j}{hP_j + hR_j}$$

where $\hat{P}_j$ is the set consisting of the predicted class and its ancestor classes, $T_j$ is the set consisting of the true class and its ancestor classes, and $| \cdot |$ denotes the number of elements in the corresponding set. The micro-averaged hierarchical precision ($hP_{\mu}$), micro-averaged hierarchical recall ($hR_{\mu}$), and micro-averaged hierarchical $F$-measure ($hF_{\mu}$) are respectively calculated as follows:

$$hP_{\mu} = \frac{\sum_{j=1}^{m} |\hat{P}_j \cap T_j|}{\sum_{j=1}^{m} |\hat{P}_j|},$$

$$hR_{\mu} = \frac{\sum_{j=1}^{m} |\hat{P}_j \cap T_j|}{\sum_{j=1}^{m} |T_j|},$$

$$hF_{\mu} = \frac{2 \times hP_{\mu} \times hR_{\mu}}{hP_{\mu} + hR_{\mu}}$$

where the number of data instances is $m$.

4.3. Experimental results

4.3.1. Comparison

As shown in Tables 2–4, OIHCRNs showed similar but lower performance compared with the other methods in terms of $hP_{\mu}$. However, in terms of $hR_{\mu}$ and $hF_{\mu}$, OIHCRNs showed comparable and even better performance than the classification methods that are not capable of incremental class learning. If a classification model is trained on a single data instance one at a time, the newly trained data can change the trained model, corrupting previously trained information, which can lead to performance degradation. In this regard, it was expected that the compared methods without incremental class learning ability would outperform our networks. Nevertheless, our networks showed lower but similar performance with a slight decrease in performance. This verified that OIHCRNs can be used in practice. OIHCRNs also showed better performance compared with ARTMAP-HC.

In the experiments, all the used datasets already have normalized values between [0,1] and the additional normalization process is not needed for ARTMAP-HC. On the other hand, OIHCRNs still need to normalize (project) input data to classify. Therefore, ARTMAP-HC outperformed OIHCRNs for the datasets, WIPO in terms of $hP_{\mu}$, $hR_{\mu}$, and $hF_{\mu}$, ImCLEF07A in terms of $hP_{\mu}$ and $hR_{\mu}$, and ImCLEF07D in terms of $hF_{\mu}$. Nonetheless, these kind of datasets are rare in reality, and OIHCRNs showed improved performance in most cases. Moreover, for the WIPO, ImCLEF07A, and ImCLEF07D datasets with the same highest class level for all data, OIHCRNs showed the same accuracies for $hP_{\mu}$, $hR_{\mu}$, and $hF_{\mu}$, while the other methods showed the different accuracies. This verified that OIHCRNs predicted the class labels with the correct highest class levels differently from the other methods.

OIHCRN-CE showed higher $hR_{\mu}$ (see Tables 2 and 4) but lower $hP_{\mu}$ (see Tables 2 and 3) compared with OIHCRN. This is because class END which is not one of actual classes is added to determine when finishing classification. Since a given input can be classified as class END during the classification prior to the highest class level, the likelihood of classification failure increases. OIHCRN-DL showed not only higher $hR_{\mu}$ (see Tables 2 and 4) but also comparable or even higher $hP_{\mu}$ (see Tables 2 and 3), and therefore $hF_{\mu}$ (see Tables 2 and 4) was improved.

4.3.2. Incremental class learning ability

In Fig. 6, each graph depicts the change in accuracy for ARTMAP-HC (top row) or OIHCRN-DL (bottom row) on each GPCR dataset. Both ARTMAP-HC and OIHCRN-DL are able to learn new data incrementally without destabilizing a previously learned network. The results of ARTMAP-HC showed a small fluctuation within an increasing tendency in the accuracies as shown in Figs. from 6(a) to 6(d). This is because the various numbers of hierarchical labels of the data instances in the dataset affected its performance when each data instance was provided for training. OIHCRN-DL, on the other hand, showed a more reliable performance. As shown in Figs. from 6(e) to 6(h), the variance of each graph was smaller than that of ARTMAP-HC, and the three evaluation measures, $hP_{\mu}$, $hR_{\mu}$, and $hF_{\mu}$, showed similar trends and values on the same dataset. The result corresponding to the last point of each graph, which was tested after training all the data, was equal to the accuracy obtained in the previous section and the upper limit of the results of the previous points.
4.3.3. Performance analysis

Each graph in Fig. 7 depicts the distribution of the highest class levels measured across the 5 × 2-fold cross-validation. The colored bars represent the average of the highest class levels, and the error bars displayed on the colored bars measure the variation in the highest class levels. The distribution of the highest class levels for the test dataset was similar to the distribution for OHCRN-DL rather than the distribution for OHCRN, on all the datasets. OHCRN-DL improved blocking so that the highest class level of each classified result is similar to the target. From the results, it can be concluded that showed better performance in terms of the recall due to the improved blocking.

It was intended to reduce model complexity by proposing models from OHCRN to OHCRN-CE and OHCRN-DL. To confirm this, the numbers of parameters for the three models were calculated and compared. As defined in the previous sections, an input vector
The colored bars represent the average of the highest class levels, and the error bars displayed on the colored bars measure the variation in the highest class levels.

Figure 7. Distributions of the highest class levels for the test dataset (red bar), OIHCRN results (green bar), and OIHCRN-DL results (blue bar) on the GPCR protein function prediction datasets, respectively.

Table 4
Micro-averaged hierarchical recalls ($h_{p^v}$) and F-measures ($h_{p^f}$) on the protein function prediction datasets in %.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Dataset</th>
<th>$h_{p^v}$</th>
<th>$h_{p^f}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GPCR Interpro</td>
<td>GPCR Pfam</td>
<td>GPCR Prints</td>
</tr>
<tr>
<td>BR-SVM1</td>
<td>73.70</td>
<td>47.68</td>
<td>71.99</td>
</tr>
<tr>
<td>BR-SVM2</td>
<td>73.50</td>
<td>47.65</td>
<td>71.09</td>
</tr>
<tr>
<td>Clus-HMC-Ens</td>
<td>77.93</td>
<td>51.86</td>
<td>76.95</td>
</tr>
<tr>
<td>LMBNbwU</td>
<td>67.29</td>
<td>59.17</td>
<td>66.32</td>
</tr>
<tr>
<td>GMNB</td>
<td>71.33</td>
<td>57.52</td>
<td>69.42</td>
</tr>
<tr>
<td>GMNBwU</td>
<td>74.76</td>
<td>60.13</td>
<td>73.00</td>
</tr>
<tr>
<td>ARTMAP-HC</td>
<td>71.21</td>
<td>37.32</td>
<td>73.57</td>
</tr>
<tr>
<td>OIHCRN</td>
<td>76.49</td>
<td>50.94</td>
<td>75.34</td>
</tr>
<tr>
<td>OIHCRN-CE</td>
<td>77.04</td>
<td>51.21</td>
<td>75.71</td>
</tr>
<tr>
<td>OIHCRN-DL</td>
<td>76.95</td>
<td>51.62</td>
<td>75.56</td>
</tr>
</tbody>
</table>

The models have the same $O(n^2)$ model complexity, but OIHCRN-DL has fewer number of parameters, almost half compared to OIHCRN and less $\sum_{l=1}^{M_b}(N_a + 2 + N_b)$ than OIHCRN-CE. The differences would be significant as the number of input vector dimensions, the clusters created, the class label for each level, and/or the given class level increase. As shown in Fig. 8, the number of parameters for OIHCRN, OIHCRN-CE, and OINCRH-DL were calculated for a class level $l$ where the number of classes $N_l$ was assumed to be $N_l = N_i$. When $1 \leq N_a, N_i = N_b \leq 10$ (Fig. 8(a)), it was clearly shown that OIHCRN-DL has the lowest number of parameters. When $5000 \leq N_a, N_i = N_b \leq 10000$ (Fig. 8(b)), the difference between the number of parameters in OIHCRN and the number of parameters in OIHCRN-CE and OIHCRN-DL, that is, the difference in model complexity, increased significantly.
We also plotted graphs of training time spent on OIHCRN-DL training for each data instance as shown in Fig. 9. When the first data instance was provided, OIHCRN-DL had to build the entire network structure once, so it took more time than training for other data instances. As new data instances were added sequentially, the learning time slightly increased to create a new class node if the existing class matched and no appropriate node was found. Nevertheless, less than 0.03 seconds were taken for training on all datasets, except for each first data instance. OIHCRN-DL was verified to be capable of real-time learning and classification as the maximum training time for a data instance is approximately 0.03 seconds. In short, OIHCRN-DL was proven practical in terms of training time.

5. Multimedia recommendation for digital storytelling

5.1. Digital companion

A software agent, called a digital companion, which resides in the smart device was developed to provide emotional interactive services to a user [6]. The digital companion was developed to have a conversation with the user using interactively generated sentences. If the digital companion can communicate with the user using additional multimedia as well as the generated sentences, the user can understand the intention of the digital companion more effectively. In order to provide multimedia in addition to a dialogue, it is essential for the digital companion to select an appropriate multimedia associated with a certain situation for more delivery effects of digital storytelling. Hence, a multimedia recommendation system was developed to select an appropriate digital file for individual multimedia features such as images and music for a given situation.

Fig. 8. Comparison of the numbers of parameters calculated for OIHCRN, OIHCRN-CL, and OINCRH-DL for a class level l where (a) \(1 \leq N_0, N_0^l = N_0^l \leq 10\) and where (b) \(5000 \leq N_0, N_0^l = N_0^l \leq 10000\). \(N_0\) and \(N_0^l\) indicate the input vector dimension and the generated cluster number, respectively. The number of classes \(N_0^l\) is assumed to be \(N_0^l = N_0^l\).

Fig. 9. Results of verifying the practical usefulness of incremental class learning of OIHCRN-DL for the GPCR protein function prediction datasets. Time consumptions for training (s) as the number of trained data instances increased.

Fig. 10. Example of OIHCRN applied to the multimedia recommendation system for digital storytelling that recommends music related to traveling.
5.2. OIHCRN For multimedia recommendation

From the experimental results in Section 4.3.1, OIHCRN showed better $h^{PA}$ than OIHCRN-CE and OIHCRN-DL for the 4 GPCR protein function prediction datasets, which include categorical data as input attributes and consist of small number of data instances. For the application to multimedia recommendation system in this section, the input attributes are binary numbers and the number of data instances is small. In addition, for multimedia recommendation, it is important to recommend more appropriate media, so we applied OIHCRN to the system.

To recommend a media file by its type and genre, two levels of modules, each of which composed of OICRN$A$ and OICRN$C$, were built and stacked hierarchically, as shown in Fig. 10. For class level 1, a media type, in other words, whether a media is an image or music, is trained according to the given input by updating the selected cluster node connected to the input field and the weights of the map field associated to the selected cluster node. For class level 2, a specific genre of the selected media type is trained according to the appended input associated with the given input and the selected media type. As each pair in the training set is encoded, OIHCRN with 2 class level modules is trained. The output nodes representing class labels of each module in OIHCRN are labeled according to the corresponding media type and genre for class level 1 and 2, respectively, and the labels are also stored.

Before selecting a media file, the trained network is loaded. OIHCRN with 2 class levels is created and initialized to the stored weights and cluster nodes for each class level. The output nodes in the map field for class levels 1 and 2 are labeled as the stored media types and genres, respectively. From the loaded network, a digital media file is selected based on the inputs such as the keywords and context information.

5.3. System description

The overall architecture of the multimedia recommendation system for digital storytelling is described in Fig. 11. As in [12], when the keywords of the conversation and context information from the user are used as inputs, and the corresponding media is selected and provided to the user while talking with a digital companion.

OIHCRN was built and trained on a digital companion embedded in the smartphone. The training set was provided as a text file containing the pairs of the keywords, context information, and corresponding media file. The trained OIHCRN was stored as text files containing the values of weights of the network. For the given inputs, the file names of image or music was selected from the trained OIHCRN. Based on the selected media file, the corresponding image or music was provided during a dialogue on a digital companion. In the demonstration, the training set that contains 27 pairs of the keywords, context information, and corresponding media file is used. Of the 35 predefined words, the words in the generated sentence were selected as the keywords.

5.4. Demonstration

To verify the effectiveness of the proposed multimedia recommendation system on a digital companion, demonstrations of the interactions between a user and a digital companion are per-
formed. Using the dataset containing pairs of keywords, context information, and corresponding media file, OIHCRN was trained to recommend an appropriate media based on dialogue and user's context information. Then, during the conversation with the digital companion in a smartphone, the appropriate media classified by its type and its genre from the trained network, was conveyed to the user. The video clip, which contains the demonstrations of the multimedia recommendation system in the smartphone applying OIHCRN, is available at http://rit.kaist.ac.kr/home/oihcrn. Snapshots of the video clip are shown in Fig. 12. When the companion talked about a movie, the companion showed a photo related to the movie during the conversation as shown in Fig. 12(a) and (b). When the user and the digital companion talked about a trip to the beach, the companion played an music related to the beach as shown in Fig. 12(c) and (d).

6. Conclusion

This paper proposed a novel OIHCRN to enable online incremental class learning in hierarchical classification. The proposed model expands the network structure of the classification module when a new object class is given and hierarchically stacks a new classification module when data having a new class level is given. That is, incremental class learning is possible by expanding the network structure developmentally. OIHCRN is one of the first attempts to enable incremental class learning in hierarchical classification, and it showed improved performance over the previous method using CIOCR, which is superior in performance, as a base module. By the proposed process of scale-preserving projection and prior label appending, OIHCRN reflects the class dependency between class levels and simultaneously normalizes the input vector online. We introduced a technique to expand the dimensions, which allows online normalization without information from the entire dataset. Besides, the input vector is rotated at an angle proportional to the reciprocal value of the object class classified in the previous class level, so that the classification result of the previous level can be reflected in the same dimension without additional normalization. OIHCRN-CE and OIHCRN-DL, which introduced additional techniques to reduce model complexity compared to OIHCRN, were also proposed.

The experimental results verified that OIHCRN, OIHCRN-CE, and OIHCRN-DL showed similar or improved performance compared with other methods. Despite the disadvantages of learning the data given one by one and disrupting the previously learned knowledge, OIHCRNs showed similar performance, which proved that our networks were practically usable in terms of performance. Among the proposed networks, OIHCRN-DL showed the best performance and the least model complexity with the smallest number of parameters. The learning time per data instance was also measured and verified to be short enough to be used in real time. Moreover, OIHCRN was applied to a multimedia recommendation system for digital storytelling of a digital companion embedded in a smartphone. The effectiveness was validated through the demonstrations of the interactions of the digital companion with the user. Using OIHCRN, it was possible to perform hierarchical classification online in real-time interactions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2017R1A2A1A17069837).

Supplementary material


References

Ju-Youn Park received the B.S. and Ph.D. degrees in Electrical Engineering from KAIST (Korea Advanced Institute of Science and Technology), Daejeon, Republic of Korea in 2015 and 2019, respectively. She worked as a Post-doctoral Researcher with the Information and Electronics Research Institute, KAIST. She is currently a Post-doctoral Scientist with the School of Engineering and Applied Science (SEAS), the George Washington University, Washington, DC, United States. Her current research interests include machine learning, artificial intelligence and robotics.

Jong-Hwan Kim received the Ph.D. degree in Electronics Engineering from Seoul National University, Seoul, Republic of Korea, in 1987. Since 1988, he has been with the School of Electrical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Daejeon, Republic of Korea, where he is leading the Robot Intelligence Technology Laboratory as a Professor. He is currently the Dean of the College of Engineering, KAIST, and the Director of the KoYoung-KAIST AI Joint Research Center and the Machine Intelligence and Robotics MultiSponsored Research Platform. He has authored five books and five edited books, two journal special issues, and around 400 refereed papers in technical journals and conference proceedings. His research interests include intelligence technology, machine intelligence learning, ubiquitous and genetic robots, and humanoid robots. He served as an Associate Editor for the *IEEE Transactions on Evolutionary Computation* and the *IEEE Computational Intelligence Magazine*. 