Learning Spatio-temporally Invariant Representations from Video

Jae Hyun Lim, Hansol Choi, Jun-Cheol Park, Jae Young Jun and Dae-shik Kim
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, Korea
lim-0606@kaist.ac.kr

Abstract—Learning invariant representations of environments through experience has been an important area of research both in the field of machine learning as well as in computational neuroscience. In the present study, we propose a novel unsupervised method for the discovery of invariants from a single video input based on the learning of the spatio-temporal relationship of inputs. In an experiment, we tested the learning of spatio-temporal invariant features from a single video that involves rotational movements of faces of several subjects. From the results of this experiment, we demonstrate that the proposed system for the learning of invariants based on spatio-temporal continuity can be used as a compelling unsupervised method for learning invariants from an input that includes temporal information.

Keywords-component: Unsupervised learning; Invariant learning; Machine vision; Learning and adaptation

I. INTRODUCTION

Learning internal representations of environments through experiences that are both spatially and temporally invariant has been an important area of research in the fields of both machine learning and computational neuroscience. Specifically, in machine learning and computer vision, it is important to develop competent representations of the inputs that have been issued in order to construct robust computational systems under various conditions of inputs. Various studies in unsupervised feature learning have revealed that the combining of the pooling operation and the learning of lower level features allows the system to be robust to various transformations of images of objects in classification, including translation or size variation [1-3]. However, these models are limited in learning features that are invariant over spatial transformations.

Temporal proximity, which has a strong statistical structure in natural video scenes, was investigated by several groups in the context of computer vision and computational neuroscience, and the learning spatio-temporal features using a temporal proximity structure has been studied in various contexts [4-9]. Cadieu and Olshausen suggested a method of learning sparse complex bases of which the amplitude factor is further learned to be ‘form selective’ and the phase factor is learned to be ‘motion-selective’ [5]. Le et al. provided an unsupervised feature learning method for learning spatio-temporal features with a hierarchical architecture; they demonstrated noticeably better results for video classification compared to methods using hand-designed features [4]. Taylor et al. suggested a convolutional learning strategy with a modified gated Restricted Boltzmann Machine (GRBM) to learn spatio-temporal features from video inputs; they also demonstrated overwhelmingly better performance than that of video classification with learning of features in unsupervised manners [6]. However, the method suggested in those papers treated the input as a concatenate of two sequential input images. The strategy of enlarging and modifying the input structure has limitations in learning the various types of transformations found in the input.

Learning of a pooling structure from the temporal proximity statistics of input data has been investigated specifically by the Numenta Corporation, which developed the concept of Hierarchical Temporal Memory (HTM) [10-12]. HTM involves a hierarchical layer structure and an important temporal-based aspect to each layer based on the concepts described in earlier research [12]. Incorporating with temporal information learned via the scanning of the input and the hierarchical layer structure, this method achieves robustness in diverse types of transformation of inputs. This is particularly true for visual information. The learning probabilistic appearance manifold suggested by Lee et al. is another notable example of the application of temporal information for a classification, specifically for video face recognition [13, 14]. This method includes the learning of low-dimensional manifolds for different subsets of inputs, each of which includes a set of faces in different orientations, as well as the learning of transition probabilities across these subsets. However, these methods are a type of supervised learning that requires the teaching of inputs for learning. Furthermore, related methods in [13, 14] are only applicable to the classification of an image within a video frame.

In contrast to these approaches, the current study suggests an unsupervised method to use temporal proximity across features to build a set of groups that consists of low level features, and to use this group structure for a pooling operation. This way of constructing the pooling structure can allow a learning system to learn adaptively the invariant structure of features from various transformations that appear in the input. Thus, what we accomplish in this study is to make the system explicitly learn, temporally as well as spatially, the invariant structures of inputs that can arise under different conditions. The prominent characteristic of this method is its use of
temporal information to cover diverse variations of instances of invariants with unsupervised learning.

II. METHODS

The main idea of our method is to use temporal proximity between the features that appear in an input of video data as a measure of temporal similarity, and to build a set of groups based on that proximity for the pooling operation. The building of a pooling structure of low level features based on the temporal information allows the suggested system to learn the invariant structure over various transformations that occur in the inputs in an unsupervised manner. To be specific, in our system’s architecture and task, we assumed that the system obtained its input as a series of images in the form of a video stream during training. The overall procedure starts with the learning of invariant features in the sense of spatial similarities in the input space. This step is referred to as spatial invariant learning. To that end, the system progresses with so-called temporal invariant learning, measuring temporal similarities over the features and learning the group structure of the features.

A. Spatial Invariant Learning

For the system to learn spatial invariants, we employed a simple version of a self-organizing feature map (SOFM) or a self-organizing map (SOM), which is also known as a ‘Kohonen map’ [15, 16]. The main purpose of the SOFM is to find significant patterns from the inputs in an unsupervised manner; the features learned through SOM provide explicit codes reflecting the input statistics. Among various implementations of the SOFM, we used a competitive learning type of SOFM. The training data set is a set of images, each of which is the frame of a single input video stream and has a size of 32 by 32 pixels.

We use $m \times n$ neurons or nodes in the SOFM. This provides $m \times n$ features as a result of the training process; each feature is located at the center of an $\mathbb{R}^{32 \times 32}$ space with respect to the vector representation of the input data. The features learned in this step are called spatial features.

At the following stage, the system is going to learn the group structures of the spatial features. The elements of the groups are the spatial features; the group structures are used in the pooling operation. For this, we assumed that if system receives an input image at time $t$, the system finds the only spatial feature that shows the best fit to the input image.

B. Temporal Invariant Learning

The main idea of this step is to build a set of groups, each of which consists of temporally similar spatial features, and to consider each group as of one invariant, allowing the system to recognize only which group is fired by the pooling operation of the group structure. (1) To construct the temporal groups by using temporal similarities across features, we measure the temporal proximity over the features and construct a temporal similarity matrix to represent the proximities. (2) Then, with the matrix, we cluster the spatial features. (3) As a result of this learning, the system obtains the group structures in which each group is one class of invariant possessing the transformation structure of the input images that appear in the video stream.

Thus, we define this temporal invariant feature as a group of spatial features that lie on closed positions on the time dimension. In other words, we consider a set of spatial features that occur continuously and frequently together as instances of a certain invariant feature. Algorithmic implementation is as follows: 1) temporal similarity measures, which are performed to measure the temporal proximities between the spatial features along the temporal dimension; 2) temporal grouping, which is performed to cluster spatial features with the similarity information.

1) Temporal Similarity Measure: The first step in the generation of temporally invariant features is to quantify how the spatial features are similar to each other along the temporal domain. To do this, we simply measure the temporal proximities across the features, measuring how each spatial feature comes together in the system within a certain size of time window. We use the ‘frequency’ of this coming together of the features as a measure for this. We set the size of the time window at $\tau$ and let the time window shift by 1 frame per each time. In other words, if different spatial features $s_i$ and $s_j$ in the SOM $S = \{s_1, s_2, \ldots, s_{num}\}$ come within a time window $\tau$, then, equal to the $i$ and $j$ are the indices of the most fired spatial features, once at $t$, $t+1$, ..., $t-\tau$, then, we add the frequency of the proximity of those two features $s_i$ and $s_j$. Then, we represent these frequencies in an $mn \times mn$ matrix. We refer to the frequencies as temporal similarities and to the matrix as a temporal similarity matrix $T$. Furthermore, we forcefully make the temporal similarity matrix symmetric; thus, the order of the indices of the spatial features is not considered when we measure the temporal similarities.

Given spatial features $S = \{s_1, s_2, \ldots, s_{num}\}$ where $s_i \in \mathbb{R}^{1024 \times 1}$

$$i = \arg \min_j \|x_i - s_j\| \quad \text{where} \quad x_i \in \mathbb{R}^{1024 \times 1}$$

(1)

$$\text{if} \; i \neq j, \; T_{i,j} \leftarrow T_{i,j} + 1$$

(2)

where $i, j$ are indices of spatial features fired from $t - \tau$ to $t$

2) Temporal grouping: In this step, we cluster the spatial features using the temporal similarities by developing the clusters to predict the temporal dynamics of the input images based on the information of the clusters. The suggested method of clustering the temporal groups consists mainly of two stages: 1) modifying the temporal similarity matrix by estimating the indirect similarities between the spatial features along the temporal dimension, and 2) clustering the features using the prediction-error form of learning method. Overall procedures can be easily understood by treating the temporal similarity matrix as an adjacency matrix of a graph.
We first modify the temporal similarity graph for the purpose of erasing the temporal similarities between the spatial features that arose accidently together and extend the temporal similarities across the spatial patterns that are indirectly but strongly similar. During the initial stages of modifying the temporal similarity graph, we eliminate the weights of the edges that are less similar by using the following procedures: 1) we find the k-nearest neighbors that have the greatest temporal similarity values within the temporal similarity graph, after which we 2) eliminate other elements within the graph and 3) convert each temporal similarity value into an inverse value, excluding the zero values. For this modified graph, we call this as modified graph $T'$. We then find the shortest paths across all combinations of spatial features within the modified graph $T'$. This gives us global neighboring structures based on ‘neighbors of neighbor' relationships across the spatial patterns. From this graph, we retain only the edges that have sufficiently small weights and erase the others so that the graph contains only edges that are considered similar. As the term ‘sufficient’ is vague, however, the implementation of this step can vary. The first method can be used to find the k-nearest neighbors within the graph and to eliminate other elements, as was done in an earlier step. Another approach involves eliminating the weights of the edges that have values less than a certain threshold $\epsilon$. At this point, from a graph that has only sufficiently small weights of edges that are inversely proportional to the temporal similarities, we again inverse each weight value of the edges. This graph is referred to as the ‘extended temporal similarity graph $T''$’. The detailed algorithmic procedure is shown in Table 1.

Next, we set the initial condition for clustering the spatial features based on $T''$. Each spatial feature starts with a consideration of itself as its own group, separated from all other features; hence, we have $mn$ classes as an initial setting. Here, we use a neural network-like form to represent the group information. Thus, the system has $mn \times mn$ weight matrix $W$. Each spatial feature has a $mn \times 1$ weight vector representing the group information. The initial value of the weight vector for each spatial feature has only one element, having a one-to-one correspondence with the clustering result, while the other elements are zeros. In other words, weight matrix $W$, representing the group information, is set as an identity matrix. After clustering the spatial features, weight matrix $W$ has only a small number of row vectors, indicating the clusters that are temporally invariant.

$$\text{Initialize } W = \begin{bmatrix} w_1 & w_2 & \ldots & w_{mn} \end{bmatrix}$$

$$W = I$$

where $I$ is $mn$ by $mn$ identity matrix

With the extended temporal similarity matrix and the initial condition, we now let the system learn the data to cluster the spatial features. The group information is represented by weight $W$. When a certain $i$-th spatial feature is fired by the current input at time $t$, this can be represented as a vector $x$, having a size of $mn \times 1$ and only one nonzero value $x_i$. This $i$-th spatial feature fires its neighbors in the ‘extended temporal similarity matrix’. Fired neighbors can be represented by multiplication between the extended temporal similarity matrix $T''$ and the impending input $x$. We then sum up the fired neighbor with $x_i$ and refer this to $x''$. At this point, we assume that we have the most probable temporal group, which is a node of the output layer, fired by the $j$-th spatial feature, which was fired by the input image frame at time $t-1$. Thus, the most probable temporal group is fired when the group has the largest value in the weight vector. The most probable temporal group, which is the $l$-th cluster in this case and was determined by the input at time $t-1$, predicts that its group elements have to be fired. This can be represented as the weights of the $l$-th cluster $w_l$, where $w_l$ is a vector containing the group information from all inputs to the $l$-th cluster. This is referred to as the prediction. Next, the weights $W$, or the group probabilities of each node in the output layer, are updated along with the difference between $x''$ and the prediction $w_l$. With a decaying term via this updating method, the system can improve the quality of its temporal grouping or clustering in the sense that it now has a

<table>
<thead>
<tr>
<th>Step</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct modified graph</td>
<td>Define the graph $T'$ over all $mn$ number of spatial invariants by connecting spatial patterns $i$ and $j$ if $i$ is one of the $k$ nearest neighbors of $j$. Set edge weights $T_{ij}'$ equal $1/T_{ij}$, if $T_{ij}'$ is linked by and edge; $T_{ij}' = \infty$ otherwise. With these initial values, compute shortest paths between all combinations of spatial invariants and update $T'$.</td>
</tr>
<tr>
<td>Find shortest paths</td>
<td>Initialize $T_{ij}'' = 1/T_{ij}'$ if $i$ and $j$ are linked by and edge; $T_{ij}'' = \infty$ otherwise. With these initial values, compute shortest paths between all combinations of spatial invariants and update $T''$.</td>
</tr>
<tr>
<td>Construct extended graph</td>
<td>Define the graph $T''$ over all $mn$ number of spatial invariants by connecting spatial patterns $i$ and $j$ 1) if $i$ is one of the $k$ nearest neighbors of $j$, or 2) if $T_{ij}' &lt; \epsilon$. Set edge weights $T_{ij}''$ equal $1/T_{ij}'$.</td>
</tr>
</tbody>
</table>
higher number of proper members, with inappropriate clusters annihilated.

\[ x' = T^* x + x \]
\[ \text{prediction} = w_t \text{ where } w_t = \langle W_{1,t}, W_{2,t}, \ldots, W_{mn,t} \rangle \]

\[ x^* = x' - \text{prediction} \]
\[ W(t+1) = (1 - \alpha) \cdot W(t) - \eta \cdot x^* \]

### III. EXPERIMENT

The current research targeted the invariant learning of face data under the condition of head rotation and the learning of the invariants of the faces of each subject as they appeared in the data set. The VidTIMIT dataset is used for the current study [17-19]. The VidTIMIT dataset consists of video and corresponding audio recordings of 43 subjects performing a head rotation and reciting short sentences corresponding to the TIMIT dataset. The recording was done in an office environment; the original video of each subject has a resolution of 512 \times 384 pixels. For computational efficiency, however, the current study used the video dataset of only 5 subjects in the VidTIMIT dataset and cropped the original video to 32 \times 32 pixels so that it contained only faces. More specifically, the current study constructed a single video stream containing 35 sessions, each of which contains a sequence of 100 images of a randomly selected subject performing a head rotation; thus, 3500 images are obtained in series as a training dataset. Some cropped frames from the training data are shown in Fig. 1. The procedure of the experiment is to let the proposed system learn the spatio-temporally invariants of each subject, independent of the head orientation, from a single video stream.

To analyze the performance of the suggested method, we first describe the results of the clustering as to whether the proposed method learned the invariant structure of the transformation of the images that occurred in the input video stream. We used the Fruchterman-Reingold graph visualization algorithm to depict the results [20]. Then, we examine the quality of the clusters under various conditions of the hyper parameters of the system, including time window \( \tau \), nearest neighbor size \( k \), and threshold \( \varepsilon \). Since our clustering method is neither a graph clustering nor a clustering over the input space, we designed an indirect method to determine the quality of the cluster. By treating the clustering result as a classification, we compared the results of the suggested method against the ground truth, which was built by using the conventional classification algorithm. We used SVM with linear kernels from all sets of images in the input video stream to evaluate which subject the spatial features comes from; however, we only considered the spatial features that are fired at least once for the given video stream since only these spatial features are valid in the suggested temporal clustering [21]. From this ground truth, the accuracy of the clustering result averaged over the number of subjects was measured.

### IV. RESULTS

#### A. Unsupervised Spatio-Temporal Invariant Learning

The result of temporal grouping for 100 spatial features under the conditions of time window \( \tau = 7 \), nearest neighbor size \( k = 1 \), and threshold \( \varepsilon = 1 \) is shown in Fig. 2. Each spatial feature learned from the spatial invariant learning is
Figure 3. (a) The measure of quality of clustering for the suggested method under the conditions that (a) the nearest neighbor size $k$ and the threshold $\varepsilon$ were varied with fixed size of time window $\tau = 7$, (b) the size of time window $\tau$ and the threshold $\varepsilon$ were varied with fixed nearest neighbor size $k = 1$, and that (c) the size of time window $\tau$ and the nearest neighbor size $k$ were varied with fixed threshold $\varepsilon = 1$. The results indicate that the suggested method is robust under variation of the nearest neighbor size $k$ for the vidTIMIT dataset. However, as the threshold $\varepsilon$ becomes decreased, the accuracy of the system is getting lower as in Fig. 3a and 3b. In the case of variation over the size of time window $\tau$, the system obtains more accuracy group structures for larger size of time window, but it has abrupt drop at the size of time window $\tau = 8$ as shown in (b) and (c).

represented as symbols, and the temporal similarities, which are weights of the temporal similarity graph, are represented as a gray solid line (class 1 = red-plus sign, class 2 = green-circle, class 3 = magenta-triangle, class 4 = black-cross, class 5 = blue-square). There are three main noticeable characteristics of the proposed method. First, the proposed method successfully discovered the proper number of clusters, 5, which was the number of subjects in the video stream. Second, the clusters, which are the ‘leaf-shaped’ branches in the low-dimensional visualization, contain invariant structures of spatial features that originate from the same subject, containing the head rotation transformation of faces that occurred in the input video stream. Third, the spatial features that were never activated in the input video stream were annihilated in determining the cluster.

B. Analysis of the quality of clusters under various hyper parameter conditions

To analyze the performance of the suggested method quantitatively, we examine the quality of the clusters under various conditions of hyper parameters of the system, including the variation of time window $\tau$, nearest neighbor size $k$, and threshold $\varepsilon$. First, the nearest neighbor size $k$ and the threshold $\varepsilon$ were varied with fixed size of time window $\tau = 7$; the result is shown in Fig. 3a. Second, the size of the time window $\tau$ and the threshold $\varepsilon$ were varied with the fixed nearest neighbor size $k = 1$; the result is shown in Fig. 3b. Third, the size of time window $\tau$ and the nearest neighbor size $k$ were varied with fixed the threshold $\varepsilon = 1$. As shown in Fig. 3a and 3c, the results indicate that the suggested method is robust under variation of the nearest neighbor size $k$ for the vidTIMIT dataset. However, as the threshold $\varepsilon$ decreased, the accuracy of the system is getting lower (Fig. 3a and 3b). In the case of variation over the size of time window $\tau$, the system obtains more accuracy group structures for larger size of time window, but it has abrupt drop at the size of time window $\tau = 8$ as shown in (b) and (c).

V. DISCUSSION

From the description of the proposed method with depicting it over graph visualization algorithm, the experimental results shows two characteristics: 1) the proposed algorithm properly approximated the number of subjects that appeared in the training dataset, which was 5, and 2) the algorithm learned large variations of spatial features that originated from independent subjects. The first characteristic indicates that a combination of the learning of spatio-temporal continuity and the prediction-error based temporal grouping method provides us with a compelling unsupervised learning method specifically for the problem of learning temporal information. The second characteristic implies that the learning of spatio-temporal continuity sheds light on a simple and novel method to learn structures of features of invariants in real world data even when the training data is simplified.

From the results of the performance measure for the suggested method under variations of hyper parameters, we could evaluate the role of the each hyper parameter and their effects. As shown in Fig. 3a and 3b, the accuracy of the system is getting lower under the conditions of lower threshold $\varepsilon$. For lower threshold $\varepsilon$, the modified graph $T'$ becomes sparser matrix as the system gets rid of the edges larger than $\varepsilon$. This makes the system lose the temporal proximity information, thus the system becomes less accurate on obtaining proper group structure.

However, the result shows that the system is robust over the variation of the nearest neighbor size $k$ (Fig 3a and 3c). Note that we re-estimate global temporal proximity across the spatial features based on local temporal proximity when we obtain the extended temporal similarity matrix $T'$. This re-evaluation of
temporal proximity across the spatial features makes the system less sensitive to the nearest neighboring size $k$.

In the case of variation in the size of time window $\tau$, the accuracy of the system to obtain proper group structures generally increases as the time window size lengthens. This indicates that the system has more appropriate temporal proximity information across the spatial features with increment of the size of time window. However, there exists abrupt drop of the accuracy at the time window size $\tau = 8$ both in Fig 3b and 3c. This alludes that the suggested algorithm is affected by the temporal structure of input video even though the results were obtained only from vidTIMIT dataset. For the detailed analysis, it is required to demonstrate the suggested method for various types of dataset. Nevertheless, the size of time window $\tau$ might have to be optimized according to the temporal structure of input video stream.

VI. CONCLUSION

This paper suggests an intuitive and efficient method to learn spatio-temporally invariants from a single video input in an unsupervised manner. The results show that the system successfully and effectively identified the invariants of all of the subjects that appeared in the training data set. This indicates that the learning of spatio-temporal continuity provides a compelling unsupervised framework specifically for the problem of dealing with input that involves temporal information.

REFERENCES


