Representation Learning using Graph Neural Nets: A case-study in HMMs

Rajan Kumar Soni[1][0000−0002−2754−3019], Karthick Seshadri[2][0000−0002−5658−141X], and Balaraman Ravindran[1][0000−0002−5364−7639]

[1] Robert Bosch Centre for Data Science and Artificial Intelligence, Indian Institute of Technology, Madras, Chennai, India
[2] National Institute of Technology, Andhra Pradesh, India
ravi@cse.iitm.ac.in, cs18s038@smail.iitm.ac.in,
karthick.seshadri@nitandhra.ac.in

Abstract. Hidden Markov models (HMMs) belong to the class of double embedded stochastic models which were originally leveraged for speech recognition and synthesis. HMMs subsequently became a generic sequence model across multiple domains like NLP, bio-informatics and thermodynamics to name a few. Literature has several heuristic metrics to compare two HMMs by factoring in their structure and emission probability distributions in HMM nodes. However, typical structure-based metrics overlook the similarity between HMMs having different structures yet similar behavior and typical behavior-based metrics rely on the representativeness of the reference sequence used for assessing the similarity in behavior. Further, little exploration has taken place in leveraging the recent advancements in deep graph neural networks for learning effective representations for HMMs. In this paper, we propose first-of-their-kind deep neural network based approaches based on graph variational autoencoder and a diffpooling based graph convolutional network to learn embeddings for HMMs and evaluate the validity of the embeddings based on subsequent clustering and classification tasks.

Keywords: Deep metric learning · Graph Neural Networks· Hidden Markov Models · Task agnostic embeddings · Graph variational autoencoders · Diff-pooling based graph convolutional networks.

1 Introduction

Hidden markov models are well known for their role as an enabler in different real word applications, such as in customer relationship, molecular biology, body posture identification, credit card fraud detection and speech technology. Typically, the following three standard problems [12] are considered to be of interest in HMMs: (i) Computing the likelihood of generating a sequence of observations, (ii) Inferring the most likely sequence of states that might have generated an observation sequence and, (iii) Computing the parameters of the HMM given an observation sequence. However an accurate estimation of the distance between HMMs is important for the performance of several descriptive, predictive and prescriptive HMM-based models and hence the problem of learning embeddings for HMMs in a metric space has become a problem of interest.

Some of the earlier attempts to find a good metric are based on the following: (i) co-emission probabilities [10], (ii) Monte Carlo approximation for measuring entropy divergence [3] and the widely used Kullback–Leibler (KL) divergence [6], (iv) graph matching [13], (v) Bayes probability of error [1], (vi) BP metric [11], based on stationary cumulative probability distribution function [15] and (vii) system statistics [8].

Typically, approaches cited above perform well only if models of similar structure exhibit similar behavior. The structure-based metrics have an inherent lapse of not accounting for cases where the HMMs have different structures yet similar behavior. Metrics based on graph-matching will become intractable and impractical as the number of nodes and edges in the HMM graph increases [9]. Behavior based metrics are influenced by the representativeness of the reference sequence used for gauging the similarity or difference between HMMs. Various graph network models in the deep learning literature have been shown to effectively infer feature representations to encode the key properties of graphs. Adoption of these models as such for HMMs is infeasible, as HMMs are rich graphs that encode time varying behavior of datasets. Further, to the best of our knowledge little exploration has been done in studying the applicability of these recently developed deep graph neural network models in the context of learning representations for HMMs. *As a first-of-its-kind attempt*, we propose a graph variational autoencoder (GVAE) based task agnostic model and a diffpoolng based graph convolutional neural network (GCN) model to learn embeddings for HMMs. The following are the key contributions of this paper:

(i) We propose and evaluate a task-agnostic model using Graph variational autoencoders and a diffpooling-based GCN model in an attempt to effectively encode both the structure and behavior of HMMs in the embeddings learnt.
(ii) We apply the learnt embeddings in the context of a representative complete-linkage and a single-linkage clustering algorithm to showcase their validity. We have also analyzed the efficacy of the embeddings learnt in the context of a classification task.

The rest of the paper is organized as follows: Section 2 describes the models and their architecture we experimented with, to propose the learned metric. Section 3 describes the experimental setup, dataset, packages used and also discusses the evaluation methods and results with respect to performance metrics. Section 4 has our concluding remarks along with some pointers for further research.

2 Methodology

This section outlines our approach to learn task agnostic embeddings for HMMs through graph variational autoencoders, and by leveraging a class-aware representation learning for HMMs using a diffpooling based network.

A hidden markov model [12] is represented as $H = (\pi, A, B, n, C)$, where π

represents the prior probability distribution over states; n is the number of states; A represents the transition matrix; B represents the matrix of emission distribution parameters where each row is formed by concatenating mean vector and linearized diagonal co-variance matrix of the corresponding state in H; C represents either an alphabet of symbols to be emitted or a continuous space of observations depending upon whether the support of the observation sequence is continuous or discrete. As we have experimented with audio datasets, we have assumed that the emission distribution in a state i of H follows a multivariate normal distribution with the probability density for the observation x of dimensionality d, in state i given by $\frac{1}{(2\pi)^{d/2}}|\Sigma_i|^{-1/2}\exp\left\{-\frac{1}{2}(x-\mu_i)\Sigma_i^{-1}(x-\mu_i)^T\right\}$, where Σ_i is the diagonal co-variance matrix and μ_i is the mean vector. The HMMs considered in this paper are ergodic in which transitions are permitted from any state to any other state.

2.1 Graph variational Autoencoder based task-agnostic embeddings

Typically, the embeddings learnt through autoencoders are optimized just with respect to the reconstruction loss and do not have any regularization in the latent space to encode the generative ability of the HMMs. Further the order in which the HMM nodes are linearized to construct the feature vector, impacts the performance of the autoencoders. To address these demerits, we have adapted the graph encoder model proposed by Kipf and Welling [7] to build a graph variational autoencoder (GVAE) for H. The proposed GVAE converts H into a latent lower-dimensional embedding ℓ. The feature vector of a state i in H (denoted by B_i) has a dimensionality of $2d$ to accommodate the concatenated parameters of the emission distribution of the state namely μ_i and the linearized form of diagonal σ_i^2. The GVAE infers a latent vector ℓ_i of dimensionality d_r for each state i in H, such that $d_r << 2d$.

The GVAE uses a GCN for inferring $\ell = [\ell_1, \ell_2, \ell_3, ..., \ell_n]$, which is a matrix of size $n \times d_r$. Assuming that, the likelihood of ℓ_i is independent of that of the other states given A and B, we get,

$$\hat{p}(\ell \mid A, B) = \prod_{i=1}^{n} \hat{p}\left(\ell_i \mid A, B\right) \tag{1}$$

As it is intractable to compute $\hat{p}(A, B)$ by marginalizing over all possible distributions for ℓ, we assume that $\hat{p}\left(\ell_i \mid A, B\right) \sim N(\hat{\mu_i}, \hat{\sigma}_i^2)$ for some $\hat{\mu_i} \in \mathbb{R}^{d_r}$ and $\hat{\sigma}_i^2 \in \mathbb{R}^{d_r}$. Parameters of the GCN are learnt using the stochastic gradient descent to optimize the KL-divergence between the inferred distribution $\hat{p}(.)$ and the ground truth $p(.)$. The optimization objective is given as 'θ' in equation 2.

$$\theta = -E_{\hat{p}(\ell|A_i B)}[\log p(A \mid \ell) + KL[\hat{p}(\ell \mid A, B) \| p(\ell))]] \tag{2}$$

p(ℓ) denotes the prior distribution in which each ℓ_i is independently sampled from $N(0, 1)$. Back propagation is used to adjust the parameters of GCN based on the error observed in the output layer. To obtain an embedding for H, we input A and B matrices to the GCN and in the inferred output matrix 'ℓ', we treat each row as the embedding of the corresponding state in H. The GCN used as the encoder model is two layered as shown in Figure 1. The transformation performed by the GCN is given by $A \times \delta(ABW^{(1)}) \times W^{(2)}$, where $W^{(1)}$ are the parameters or weights of the GCN in the first layer. $W^{(1)}$ is the shared weight matrix of $L^{(1)}$ which is shared by the two sub layers $L_1^{(2)}$ and $L_2^{(2)}$. $L_1^{(2)}$ and $L_2^{(2)}$ separately infers $\hat{\mu}_i$ and $\hat{\sigma}_i^2$. δ is the *Relu* activation given by $max(ABW^{(1)}, 0)$. The decoder that generates the state transition matrix $\hat{A}$ given the latent parameters, is modeled using the dot product between the corresponding latent vectors as given in Figure 1.

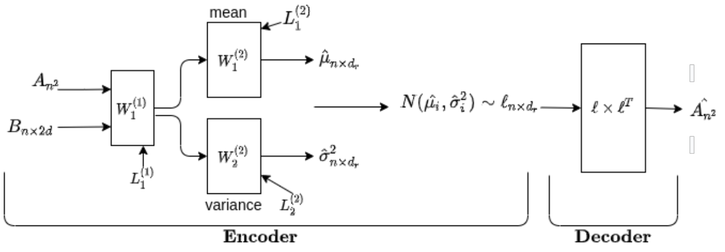

Fig. 1. GVAE model

The conditional likelihood of $A_i (\forall i \in [1, n])$ is assumed to follow a Dirichlet distribution as in equation 3. The distance between two HMMs can be computed using the Graph variational autoencoder as outlined in the Algorithm 1.

$$p\left(A_i \mid \ell_i, \ell\right) = \operatorname{Dir}\left(A_i \mid \operatorname{softmax}\left(\hat{A}_i\right)\right) \tag{3}$$

Algorithm 1 Embedding HMMs using Graph variational autoencoder

```
Input: $H_1$, $H_2$.
$X_1 = Graph\_VAE(A_1, B_1)$, $X_2 = Graph\_VAE(A_2, B_2)$
$distance = 0$
for each $v_i \in H_1$ do
    $v_j = \underset{v_k \in H_2}{\arg\min}\ euclidean\_distance(X_1(v_i), X_2(v_k))$
    distance $+=$ $euclidean\_distance(X_1(v_i), X_2(v_j))$
end for
return distance
```

2.2 Hierarchical embedding for HMMs using diffpooling

One of the demerits of the GVAE based approach is that the HMM representations learnt are flat; It has no innate provision to encode the hierarchical structures which are typically prevalent in HMMs. The overall behavior exhibited by a HMM may be viewed as the aggregation of local behaviors exhibited by a set of closely knit clusters of states. The idea proposed by [14] for learning structural graph representations has been adapted by us to encode both the hierarchical spectral-structures and behavior exhibited by the spectral-substructures in HMMs as illustrated in Figure 2. The hierarchical view captures detailed prominent localized features responsible for the overall behavior of a HMM, as opposed to Autoencoder based approaches that at one stroke pool the individual node embeddings of a HMM to emit a global embedding, thereby losing important hierarchical local patterns.

Each layer $L(i)$ in the diffpooling network corresponds to a pair of GNNs $GNN_M^{(i)}$ and $GNN_E^{(i)}$, having n_{i-1} input nodes and n_i output nodes followed by a diffpooling sub layer named $DP(i)$. In the layer L(i), n_{i-1} corresponds to the number of nodes in the input graph $G^{(i-1)}$ and n_i indicates the number of clusters in the output or nodes in the coarsened graph $G^{(i)}$ at level i.

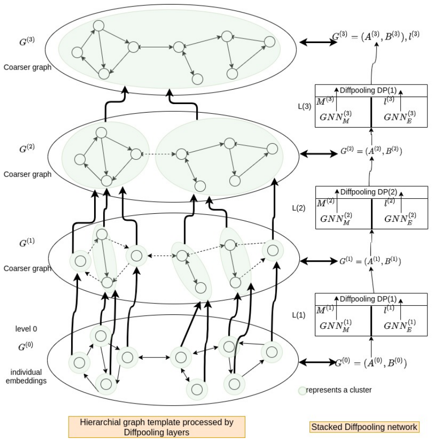

Fig. 2. Learning hierarchical HMM embedding -an illustration

Each $GNN_M^{(i)}$ maps nodes in $G^{(i-1)}$ to their degree of association with respect to each output cluster node in $G^{(i)}$, this mapping is done using the node embeddings in $G^{(i-1)}$. $GNN_M^{(i)}$ outputs the mapping of the node in $G^{(i-1)}$ to a set of nodes in $G^{(i)}$ in the form of a $n_{i-1} \times n_i$ matrix $M^{(i)}$, where each row j in $M^{(i)}$ corresponds to the strength of the mapping of j to each of the nodes in $G^{(i)}$ as specified in equation 4.

$$M^{(i)} = \mathrm{softmax}\left(\mathrm{GNN}_M^{(i)}\left(A^{(i-1)}, B^{(i-1)}\right)\right). \tag{4}$$

Similarly each of the $GNN_E^{(i)}$ accepts the adjacency and emission matrices of $G^{(i-1)}$ and generates the embedding $\ell^{(i)}$ of each cluster in $G^{(i)}$ as outlined in equation 5.

$$\ell^{(i)} = \mathrm{GNN}_E^{(i)} \left(A^{(i-1)}, B^{(i-1)} \right) \tag{5}$$

For each layer $L(i)$, $A^{(i-1)}$ and $B^{(i-1)}$ correspond to the transition and feature probability matrices of the HMM respectively. The number of output nodes in $GNN_M^{(i)}$ and $GNN_E^{(i)}$ is a hyper parameter that corresponds to the maximum number of clusters to be inferred in $L(i)$. The diffpooling sub-layer in $L(i)$ takes as inputs the embeddings of the nodes in $\ell^{(i)}$, the mapping matrix $M^{(i)}$ and $A^{(i-1)}$ to generate the feature matrix of the coarsened graph $G^{(i)}$ i.e. $(B^{(i)})$ and the adjacency matrix of $G^{(i)}$ i.e. $(A^{(i)})$ of sizes $n_i \times d$ and $n_i \times n_i$ respectively. The feature matrix $B^{(i)}$ is computed as the weighted aggregation of embeddings in $\ell^{(i)}$, where the weight of an embedding is the strength of association of the node to the output clusters is computed as $B^{(i)} = {M^{(i)}}^{\top} \ell^{(i)}$. The strength of the association between pairs of clusters/nodes in $G^{(i)}$ is computed as $A^{(i)} = {M^{(i)}}^{\top} A^{(i-1)} M^{(i)}$.

If there are m diffpooling layers then the embedding $l^{(m)}$ is considered as the final embedding of the HMM. The end-to-end training of the diffpooling network has been done using stochastic gradient descent. The distance between two HMMs can be computed as the distance between the hierarchical embeddings of the HMMs obtained by a diffpooling network. The embeddings compared are the ones emitted from the last diffpooling layer.

3 Experimental setup and Performance evaluation

The dataset used for our experiments is the open source Free Spoken Digi Dataset (FSDD). The dataset has $2K$ audio files created by four speakers uttering digits from 0 to 9. We have used google colab for performing our experiments. The key libraries used are pytorch, numpy and scipy.

To validate the embeddings generated, we have performed two extrinsic tasks namely, clustering and classification using the embeddings. For the clustering task we have experimented using a complete-linkage based agglomerative clustering and a single-linkage based Minimum Spanning Tree (MST) clustering algorithm [4]. The validity of the clusters is assessed through Cluster Purity (CP), Normalized Mutual Information (NMI) and Rand Index (RI). Similarly, classification accuracy is used as the metric for assessing the classification task. Throughout this section, $M1$ to $M8$ denote the metrics as mapped below: (i) $M1$: Cross Likelihood based metric [12], (ii) $M2$: State mapping based structural metric [13], (iii) $M3$: Unisequence Likelihood metric [3], (iv) $M4$: Matrix Factorization based linear embedding Metric [2] (v) $M5$: Hybrid metric based on both structure and behavior [12,13], (vi) $M6$: Autoencoder based metric [5] (vii) $M7$: Graph Autoencoder based metric, and (vii) $M8$: Diffpooling based metric.

Training Set Size vs. Cluster quality: The objective of this experiment is to determine the impact of the training set size on the clustering performance. We trained a set of HMMs each with ten audio files of the same digit, sampled uniformly at random with replacement from FSDD. Let S be the set of HMMs trained which has an equal representation of all the digits in $[0, 9]$. We have split S into S_{train} and S_{test} such that $|S_{train}|$: $|S_{test}| = 4 : 1$. We trained the Graph Autoencoder and Diffpooling based models using S_{train} and tested the models using the tuples in S_{test}. From the plots in Figure 3, it can be observed that the graph autoencoder based embeddings performed better than the autoencoder based embeddings and matrix factorization based linear embeddings. The behavioral baselines perform better than the structural baseline. Further the M7 and M8 embeddings required lesser number of samples in the training set to achieve a better accuracy in the complete-linkage clustering as compared to the MST-based single-linkage clustering. Diffpooling exhibited a robust performance, which is better than the rest of the models. The performance of $M7$ degrades as the training set size increases as opposed to $M8$; this is due to fact that the embeddings learnt by GVAE for different classes are not well-separated as compared to that of Diffpooling network; Further, as the dataset size grows, the discriminative ability of the decision boundary learnt also diminishes.

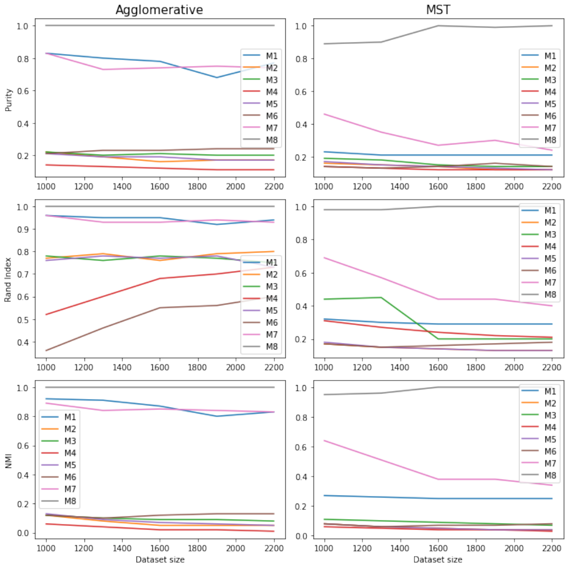

Fig. 3. Training set size vs. Cluster quality

Variance in the number of HMM states vs. Cluster quality: This experiment is designed to assess the ability of the metric to recognize the similarity between two HMMs having different number of states but similar behavior. As shown in Figure 4, compared to the other metrics, the GVAE based metric exhibited a robust performance as the variance in n increases. This asserts that the embeddings learnt by GVAEs have the ability to recognize structurally different yet behaviorally similar HMMs.

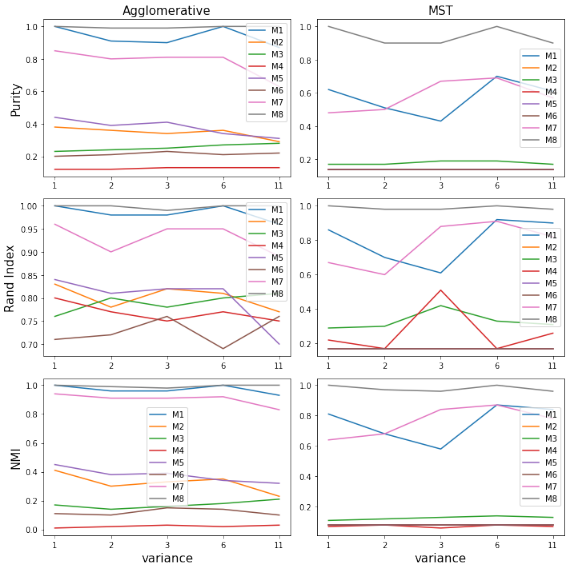

Fig. 4. Variance in the No. of HMM states vs. Cluster quality

For the classification task, the autoencoder, GVAE and GCN based embeddings achieved an accuracy of 0.45, 0.96 and 1.00 respectively for the FSDD dataset. The performance of diffpooling based GCN model is not surprising as the representations learnt are label-aware.

4 Conclusion

In this paper we have introduced a novel method for learning embeddings for HMMs that uses a graph variational autoencoder (GVAE) and a GCN to learn flat and hierarchical embeddings respectively. The embeddings learnt are effective even in tasks where the dataset contains structurally dissimilar yet behaviorally similar HMMs. This is due to the regularized, behavior-preserving and generative latent space learnt by these models. While other metrics falter when used with single-linkage clustering tasks, diffpooling exhibits a robust performance irrespective of the clustering algorithm employed. Future research will include testing the efficacy of these models on non-ergodic HMM variants.

References

1. Bahlmann, C., Burkhardt, H.: Measuring hmm similarity with the bayes probability of error and its application to online handwriting recognition. In: Proceedings of Sixth International Conference on Document Analysis and Recognition. pp. 406–411 (Sep 2001)
2. Baker, K.: Singular value decomposition tutorial. The Ohio State University **24** (2005)
3. Falkhausen, M., Reininger, H., Wolf, D.: Calculation of distance measures between hidden markov models. In: EUROSPEECH. pp. 1487–1490 (1995)
4. Jana, P.K., Naik, A.: An efficient minimum spanning tree based clustering algorithm. In: 2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS). pp. 1–5 (2009)
5. Jordan, J.: Introduction to autoencoders. Jeremy Jordan, Mar (2018)
6. Juang, B.H., Rabiner, L.R.: A probabilistic distance measure for hidden markov models. AT & T Technical Journal **64**(2), 391–408 (1985)
7. Kipf Thomas, N., Max, W.: Variational graph auto-encoders. In: Conference on Neural Information Processing Systems (NeurIPS) Workshop on Bayesian Deep Learning (2016)
8. Lu, C., Schwier, J.M., Craven, R.M., Yu, L., Brooks, R.R., Griffin, C.: A normalized statistical metric space for hidden markov models. IEEE transactions on cybernetics **43**(3), 806–819 (2013)
9. Lubiw, A.: Some np-complete problems similar to graph isomorphism. SIAM Journal on Computing **10**(1), 11–21 (1981)
10. Lyngsø, R., Pedersen, C., Nielsen, H.: Metrics and similarity measures for hidden markov models. Proceedings. International Conference on Intelligent Systems for Molecular Biology p. 178—186 (1999)
11. Panuccio, A., Bicego, M., Murino, V.: A hidden markov model-based approach to sequential data clustering. In: Caelli, T., Amin, A., Duin, R.P.W., de Ridder, D., Kamel, M. (eds.) Structural, Syntactic, and Statistical Pattern Recognition. pp. 734–743. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
12. Rabiner, L., Juang, B.: An introduction to hidden markov models. IEEE ASSP Magazine **3**(1), 4–16 (1986)
13. Sahraeian, S.M.E., Yoon, B.J.: A novel low-complexity hmm similarity measure. IEEE Signal Processing Letters **18**(2), 87–90 (2011)
14. Ying, R., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 4805–4815. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
15. Zeng, J., Duan, J., Wu, C.: A new distance measure for hidden markov models. Expert systems with applications **37**(2), 1550–1555 (2010)