ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Mixture of Hidden Markov Models as Tree Encoder

Davide Bacciu* and Daniele Castellana

Dipartimento di Informatica - Università di Pisa - Italy

Abstract. The paper introduces a new probabilistic tree encoder based on a mixture of Bottom-up Hidden Tree Markov Models. The ability to recognise similar structures in data is experimentally assessed both in clusterization and classification tasks. The results of these preliminary experiments suggest that the model can be successfully used to compress the tree structural and label patterns in a vectorial representation.

1 Introduction

The ability to learn a rich and compact encoding of structured information into a fixed-size vectorial representation is a key enabler in structured data processing tasks. One of the most challenging problems in the structured domain is that of learning a non-isomorphic transduction from non aligned structured data [1]. This defines as a generalization of supervised learning where both inputs and outputs are structured samples of unconstrained topology. In recent years, various solutions have been proposed to address the transduction problem in the sequential domain using deep learning techniques [2]. These solutions are often based on an *encoder-decoder* scheme, where the *encoder* model compresses information on the full input sequence in a fixed-length representation while the *decoder* generates the output sequence by sampling conditioned on the compressed representation produced by the encoder [2]. Related approaches are being developed also for tree structured data [3], altough none of them tackles the most general challenge of learning non-isomorph tree transduction.

This paper introduces a new probabilistic tree encoder realized as a mixture of *Bottom-up Hidden Tree Markov Models* (BHTMMs) [4]. The proposed approach exploits the mixture assumption to allow the probabilistic model to better capture varied tree structures with respect to a single or to multiple independent BHTMMs. We show how the proposed mixture of hidden trees can be used both for supervised tasks, as in the original BHTMM, as well as for unsupervised tasks, such as structure clusterization. A preliminary analysis is also conducted to asses the effectiveness of the model as an encoder, by using it to generate a compressed representation of an input tree that is then fed to a neural layer to perform tree classification.

*This work is funded by Italian Ministry of Education, University, and Research (MIUR) under project SIR 2014 LIST-IT (grant n. RBSI14STDE)

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

2 Mixture of BHTMM

Let us consider a dataset $\mathcal{D} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ of N i.i.d. structured samples, where $\mathbf{x}^n$ is a labelled and rooted tree with maximum out-degree L. Each vertex u in the tree is associated with a label x_u which we consider with values over the finite set $[1, \ldots, M]$. The goal is to learn a generative model $P(\mathbf{x}^n)$ which maximise the likelihood of the dataset $\mathcal{D}$. In this paper, we model the generative process as mixture of T BHTMM [4].

The BHTMM defines a generative process for a tree $\mathbf{x}^n$ which goes from the leaves to the root. As in standard HMM, the process is guided by the evolution of an hidden dynamics. In particular, an hidden state variable Q_u, with values over a finite set $[1, \ldots, C]$, is associated to each node u and emits the associated label x_u. The hidden state variables are linked together, reproducing the same tree structure of the visible labels. The links go from child nodes to their parent, assuming that the sate of an hidden node depends on the joint configuration of its hidden child nodes. Usually, the computation of this *state-transition* distribution is impractical, due to the exponential growth of the joint configuration w.r.t. the maximum out-degree L. The BHTMM factorises such joint state distribution as a mixture of pairwise child-to-parent transitions: this approximation is called *switching parents* (SP) [4]. Also, the model assume that an hidden state Q_u variable contains enough information to generate the visible label associated x_u.

In order to combine together different BHTMMs in such a way that they exchange information, we introduce a mixture variable K which indexes the different BHTMMs. Therefore, the likelihood of a tree $\mathbf{x}^n$ is given by:

$$P(\mathbf{x}^n) = \sum_{t=1}^{T} P(K = t) P(\mathbf{x}^n \mid K = t) \tag{1}$$

where the value $P(\mathbf{x}^n \mid K = t)$ indicates the likelihood of the sample $\mathbf{x}^n$ according to the t-th BHTMM and the value $P(K = t)$ is the weight associated to the t-th mixture's component. The value of $P(\mathbf{x}^n \mid K = t)$ can be derived summing the complete likelihood $P(\mathbf{x}^n, \mathbf{Q}^n \mid K = t)$ over the hidden state variables. The complete likelihood is obtained using the conditional independence assumptions introduced by BHTMM:

$$\begin{aligned} P(\mathbf{x}^n,\mathbf{Q}^n \mid K = t) = & \prod_{u \in \mathcal{LF}^n} P(Q_u \mid K = t) P(x_u^n \mid Q_u, K = t) \\ & \times \prod_{v \in \mathcal{U}^n} \sum_{l=1}^{L} P(S_v = l \mid K = t) P^l(Q_v \mid Q_{ch_l(v)}, K = t) P(x_v^n \mid Q_v, K = t) \end{aligned} \tag{2}$$

where $\mathcal{LF}^n$ is the set of leaves in the n-th tree and $P(Q_u \mid K = t)$ is the *priori distribution* defined by the t-th BHTMM on them. Similarly, $\mathcal{U}^n$ denotes the set of internal nodes in the n-th tree while $P(S_v = l \mid K = t)$ is the switching parents distribution defined by the t-th BHTMM and measures the weight of

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

the contribution of the l-th child to the state transition of node v. Finally, $P^l(Q_v \mid Q_{ch_l(v)}, K = t)$ indicates the dependency between a node and its l-th child according the t-th BHTMM. The expression derived is the same obtained in [4], apart from the conditioning over the mixture variable K. This dependency is needed to obtain T different models, each of them with different potentials.

Learning the mixture model parameters can be achieved using the EM algorithm. During the E-step, we would like to estimate the posterior distribution of the hidden variables given the visible ones. The posteriors of the BHTMM hidden variables are obtained applying the *upward-downward algorithm* following [4]. The *upward pass* is a bottom-up recursive procedure which computes recursively the following quantities:

$$\begin{aligned}\beta_u(i,t) = \; & P(Q_u = i \mid \mathbf{x}_u^n, K = t) \quad = \\ & = \frac{P(x_u^n \mid Q_u = i, K = t) \sum_{l=1}^{L} P(S_v = l \mid K = t) \beta_{u,ch_l(u)}(i,t)}{\sum_{j=1}^{C} P(x_u^n \mid Q_u = j, K = t) \sum_{l=1}^{L} P(S_v = l \mid K = t) \beta_{u,ch_l(u)}(j,t)}\end{aligned} \tag{3}$$

where $\beta_{ch_l(u)}(i,t) = \sum_{j=1}^{C} P^l(Q_u = i \mid Q_{ch_l(u)} = j, K = t)$ is an auxiliary value. The base case of the upward recursion are the leaf nodes, since the numerator in (3) simplifies to $P(x_u^n \mid Q_u = i, K = t) P(Q_u = i \mid K = t)$. The β-recursion ends when the root node is reached; moreover, the root value is used as the base case for the *downward pass*, since $\epsilon_1(i,t) = \beta_1(i,t) = P(Q_u = i \mid \mathbf{x}^n, K = t)$. Then, the posterior is computed for all the other nodes starting from the root and following the recursive rule

$$\begin{aligned}\epsilon^l_{u,ch_l(u)}(i,j,t) = P(Q_u = i, Q_{ch_l(u)} = j \mid \mathbf{x}^n, K = t) \quad = \\ = \frac{\epsilon_u(i,t) \beta_{ch_l(u)}(j,t) P(S_u = l \mid K = t) P^l(Q_u = i \mid Q_{ch_l(u)} = j, K = t)}{\sum_{l'=1}^{L} P(S_u = l' \mid K = t) \beta_{u,ch_{l'}(u)}(i,t)}.\end{aligned} \tag{4}$$

and marginalising over the parent variable. Also, we compute the posterior of the mixture variable K using the equation:

$$P(K = t \mid \mathbf{x}^n) = \frac{P(\mathbf{x}^n \mid K = t) \times P(K = t)}{P(\mathbf{x}^n)} \tag{5}$$

where $P(\mathbf{x}^n \mid K = t)$ is the likelihood computed in the *upward pass* [4] for each BHTMM and $P(K = t)$ is the prior distribution of the mixture variable.

The M-step update equations for the BHTMM parameters are the same derived in [4], despite of using posteriors computed according to the equation (4). The update equation for the mixture variable priori is obtained as follow:

$$P(K = t) = \frac{\sum_{n=1}^{N} P(K = t \mid \mathbf{x}^n)}{N}. \tag{6}$$

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

3 Experimental Analysis

First, we have performed a preliminary test on controlled data to assess wether the mixture of hidden trees (MIX-BHTMM) offers an advantage with respect to a single BHTMM in terms of identifying structural regularities. We composed a synthetic dataset containing ternary trees of 3 types, i.e. left-asymmetric, symmetric and right-asymmetric, depending on the proportion of nodes in the children subtrees. Each tree is generated by a top-down recursive procedure: for each node two distributions are sampled due to generate child nodes and their labels. Both MIX-BHTMM and BHTMM have been trained on data in an unsupervised setting. At test time, the MIX-BHTMM determines the cluster assignment by sampling the posterior of the mixture variable while the BHTMM determines it by sampling the posterior of the hidden root variable. Different configurations of both models have been validated, varying the hidden state number C for BHTMM and the number T of mixture components. Each configuration has been trained 5 times with a different random initial configuration. Clustering quality is assessed by means of the Silhouette index [5]. For our purpose, we measure the distance between two trees applying the Ruzicka distance [6] on theirs representative matrix, where a representative matrix A^n for a tree $\mathbf{x}^n$ is a matrix such that the value a^n_{ij} counts how many times the label j appears in a node in the i-th position in the $\mathbf{x}^n$ tree. The results obtained in Table 1 shows clearly the benefits of the mixture variable. The MIX-BHTMM score is always closer to the score obtained using the ground truth (i.e. assigning each tree of a different type in a different cluster) which is 0.15. Viceversa, the single BHTMM score is always near to 0.

Following, we have tested MIX-BHTMM on two data sets from the 2005 and 2006 INEX Competition [7]. The INEX05 dataset is made up of $9,361$ trees ($4,820$ in the training set and $4,811$ in the test set), 11 classes and 366 label. The maximum out degree is 32. The INEX06 dataset is made up of $12,107$ trees ($6,053$ in the training set and $6,054$ in the test set), 18 classes and 65 label. The maximum out degree is 66. For these datasets, we assess both the clustering quality as well as the ability of the MIX-BHTMM to provide an effective encoding of tree data for supervised learning. In particular, a fully connected feedforward network has been attached as read-out layer to perform tree classification from its compressed representation generated by the mixture. In our experiment, the compressed representation of an INEX tree $\mathbf{x}^n$ is given by a matrix A of size $C \times L \times T$ where the value a_{ilt} is the sum of the posterior $P(Q_u = i \mid \mathbf{x}^n, K = t)$ for all hidden variables Q_u which are in the l-th position (i.e. they are the l-th child of its parent). Note that, despite the presence of the supervised neural layer, training of the MIX-BHTMM is always performed in a fully unsupervised way. A model selection procedure using a 3-fold CV on the training data has been used to select the number of hidden neurons among $\{20, 40, 60, 80, 100\}$: results in Table 2 show the tested performance on the best configuration in validation.

From the clustering perspective, results on the INEX05 dataset (see Table 2)

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Silhouette index	$T = 3$	$T = 5$	$T = 7$
$C = 2$	**0.18** (0.05)	0.16 (0.04)	0.15 (0.03)
$C = 4$	0.15 (0.03)	0.12 (0.05)	0.14 (0.01)
$C = 6$	0.14 (0.01)	0.13 (0.01)	0.15 (0.03)
Single BHTMM	0.01 (0.08)	−0.01 (0.07)	−0.11 (0.08)

Table 1: Mean silhouette index over 5 runs (std in brackets) on a synthetic dataset. For the BHTMM, $T = C$, i.e. the number of hidden states.

highlight that the introduction of a mixture encourages to solve the task using a small number of hidden states (the best score is obtained with $C = 2$). In fact, an high number of hidden states implicates more expressive mixture components which can model multiple tree structures. As a consequence, a single mixture component could be selected, vanishing the mixture benefits. This behaviour is more evident on INEX06 where there seems to be a the lack of clear structural differences among trees. The models with $C = 8$ tend to create only one cluster and therefore the Silhoutte index is not defined.

With respect to the supervised setting, the results on INEX05 show that the state of the mixture variable provides useful information to the classifier. In fact, the classification error is lower when more mixture components are used. Nonetheless, when increasing the number of hidden states the classifier is also able to address the task due to the increased expressiveness of each mixture component. The advantages of using a mixture variable is less evident on INEX06 (Tab. 2) again due to the lack of clear structural differences among trees. The INEX benchmarks allow to confront MIX-BHTMM with two related approaches. The first is the input-output BHTMM (IO-BHTMM) [8], which has been proposed to introduce discriminative training of BHTMMs by having a single shared model for all classes (like with MIX-BHTMM), instead that a single independent BHTMM per class. The second is a generative model for tree clustering on topographic maps (GTM-SD) [9] that, like MIX-BHTMM, is fully unsupervised and can be used for classification by adding a classifier on the generated tree encodings. The lowest errors achieved by IO-BHTMM are of 9.83% for INEX05 and 72.39% for INEX06 being, in both cases, over 2% worse than those by MIX-BHTMM. The GTM-SD achieves a performance comparable to that of MIX-BHTMM, with an error of 7.52% for INEX05 and 69.76% for INEX06 [10]. However, in both cases GTM-SD uses 400 hidden states while MIX-BHTMM uses 44 and 72 hidden states for INEX05 and INEX06, respectively, denoting a better exploitation of the encoding space.

4 Conclusion

We have introduced a mixture of hidden tree models based on BHTMM. The introduction of the mixture allows to capture structural information from data more efficiently than using a single BHTMM, resulting on models of lesser complexity (in terms of total number of states) and yielding to more discriminate encodings. This is the consequence of the introduction of the mixture variable, which represents contextual information shared among all tree nodes. There is

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

INEX05			
Silhouette index	$T = 6$	$T = 11$	$T = 22$
$C = 2$	0.12 (0.01)	0.13 (0.07)	**0.20** (0.04)
$C = 4$	0.13 (0.09)	0.17 (0.02)	0.15 (0.02)
$C = 8$	0.08 (0.00)	0.11 (0.05)	0.17 (0.06)
Classification error	$T = 6$	$T = 11$	$T = 22$
$C = 2$	17.66 (1.55)	13.74 (4.76)	**7.30** (1.33)
$C = 4$	12.38 (4.35)	8.46 (4.08)	9.75 (4.70)
$C = 8$	10.53 (4.45)	7.73 (3.24)	9.05 (3.81)
INEX06			
Silhouette index	$T = 9$	$T = 18$	$T = 30$
$C = 2$	0.05 (0.00)	0.05 (0.01)	0.06 (0.00)
$C = 4$	0.08 (0.00)	0.04 (0.00)	**0.09** (0.00)
$C = 8$	–	0.06 (0.00)	–
Classification error	$T = 9$	$T = 18$	$T = 30$
$C = 2$	72.51 (1.39)	73.77 (2.45)	73.25 (1.43)
$C = 4$	69.47 (2.92)	72.26 (3.20)	70.19 (1.40)
$C = 8$	**69.17** (1.60)	69.29 (1.07)	71.63 (3.40)

Table 2: Mean scores for the INEX datasets over 5 runs (std in brackets).

no guarantee that a single BHTMM learns to model such information using only local dependencies. Also, we have shown how MIX-BHTMM can act as *encoder* merging together the posterior of all nodes due to compress tree information in a fixed representation. Given this preliminary results, we are planning to use it in the more challenging scenario of learning tree-to-tree transductions, where the encoded information provides context to generate the output tree.

References

[1] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE transactions on Neural Networks*, 9(5):768–786, 1998.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proc. of the 2014 EMNLP*, pages 1724–1734, 2014.

[3] D. Alvarez-Melis and T. S. Jaakkola. Tree-structured decoding with doubly-recurrent neural networks. *Proc. of ICLR 2017*, 2017.

[4] D. Bacciu, A. Micheli, and A. Sperduti. Compositional generative mapping for tree-structured data; part i: Bottom-up probabilistic modeling of trees. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(12):1987 –2002, 2012.

[5] E. Rendón, I. Abundez, A. Arizmendi, and E.M. Quiroz. Internal versus external cluster validation indexes. *Int. J. Comp. and Comm.*, 5(1):27–34, 2011.

[6] M.M. Deza and E. Deza. Encyclopedia of distances. In *Encyclopedia of Distances*, pages 1–583. Springer, 2009.

[7] L. Denoyer and P. Gallinari. Report on the xml mining track at inex 2005 and inex 2006: categorization and clustering of xml documents. In *ACM SIGIR Forum*, volume 41, pages 79–90. ACM, 2007.

[8] D. Bacciu, A. Micheli, and A. Sperduti. An input-output hidden Markov model for tree transductions. *Neurocomputing*, 112:34–46, 2013.

[9] D. Bacciu, A. Micheli, and A. Sperduti. Compositional generative mapping for tree-structured data - part II: Topographic projection model. *IEEE Trans. Neural Netw. Learning Syst.*, 24(2):231–247, 2013.

[10] D. Bacciu, A. Micheli, and A. Sperduti. Adaptive tree kernel by multinomial generative topographic mapping. In *Proc. of the IJCNN 2011*, pages 1651–1658. IEEE, 2011.