ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Input-Output Hidden Markov Models for Trees

Davide Bacciu[1] and Alessio Micheli[1] and Alessandro Sperduti[2] *

1- Dipartimento di Informatica - Università di Pisa - Italy
2- Dipartimento di Matematica Pura e Applicata - Università di Padova - Italy

Abstract. The paper introduces an input-driven generative model for tree-structured data that extends the bottom-up hidden tree Markov model with non-homogenous transition and emission probabilities. The advantage of introducing an input-driven dynamics in structured-data processing is experimentally investigated. The results of this preliminary analysis suggest that input-driven models can capture more discriminative structural information than non-input-driven approaches.

1 Introduction

Input/Output Hidden Markov Models (IO-HMMs) enable non-homogenous and input-driven state transition and emission dynamics within HMM models for sequential data [1]. The underlying intuition is simple, i.e. learning an input-conditional hidden process for the output sequence, and it has shown good potential for learning sequence transductions between different modalities [2]. However, its practical advantage with respect to homogenous models is still questioned [3]. In a sequence classification task, it is unclear whether the discriminative power of IO-HMM is superior to that of multiple HMMs, each modeling a single class distribution. Similarly, in regression tasks, HMMs can effectively be used to model the joint distribution of the dependent and independent variables, in place of learning the input-conditional distribution of the dependent variable as in IO-HMM [3]. We investigate the opportunity of input-driven HMMs dealing with more complex, tree-structured data. The contribution of this paper is twofold. First, we propose an *Input-Output Bottom-up Hidden Tree Markov Model* (IO-BHTMM), the first practical input-driven generative model for tree-structured data. This is mostly due to the strong computational limitations imposed by bottom-up generative modeling of trees, that have only recently been addressed [4] by means of an efficient finite mixture approximation which is also at the basis of the proposed IO-BHTMM. Secondly, we confront the proposed model with homogenous generative models in literature, to empirically assess, for the first time, if an input-driven approach is capable of capturing more discriminative structural information than its homogenous counterpart. To this end, we start our empirical analysis from the simplest non-trivial scenario, comprising the transduction of an input tree into an isomorph output within a structure classification scenario. A successful achievement on such preliminary analysis, would pave the way to a deeper assessment of input-driven generative models on more general transductions.

*This work is partially supported by the EU FP7 RUBICON project (contract n. 269914). The work by A. Sperduti was supported by the Italian Ministry of Education, University, and Research (MIUR) under Project PRIN 2009 2009LNP494 005.

2 IO-HMM for Bottom-up Tree Processing

Let us consider a dataset $\mathcal{D} = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$ of N structured samples $(\mathbf{x}^n, \mathbf{y}^n)$, where $\mathbf{x}^n$ is a labeled and rooted input tree with maximum finite out-degree L (i.e. the maximum number of children of a node). Each vertex u in the input tree is associated with a label x_u which, for the sake of this paper, is a d-dimensional discrete vector. The corresponding output $\mathbf{y}^n$ is a labeled rooted tree with the same structure as the input tree, but different discrete labels y_u associated to the vertices u. The task is to learn a conditional generative model $P(\mathbf{y}^n|\mathbf{x}^n)$ for the structured output, given input $\mathbf{x}^n$. This corresponds to learning an IO-isomorphic transduction (i.e. a relabeling of the nodes) from inputs $\mathbf{x}^n$ to the outputs $\mathbf{y}^n$. As in standard HMM, learning of $P(\mathbf{y}^n|\mathbf{x}^n)$ is achieved by introducing an hidden state variable Q_u following the same indexing as the observed nodes u, with values over the finite set $[1, \ldots, C]$.

Introducing the hidden state Q_u allows to simplify the conditional distribution $P(\mathbf{y}^n|\mathbf{x}^n)$ by using the conditional independence assumptions defined by the underlying hidden Markov model. In this paper, we use the *Bottom-up Hidden Tree Markov Model* (BHTMM) [4], that defines a generative unconditional process $P(\mathbf{x}^n)$ for a tree $\mathbf{x}^n$ that flows from the leaves to its root. In the unconditional BHTMM, the Markovian assumption affirms that Q_u summarizes enough information to emit the node labels. In our conditional scenario, the state variable emits the output label y_u conditioned on the input x_u, resulting in the conditional *emission* $P(y_u|Q_u = i, x_u)$. In the bottom-up approach, the hidden state of a node Q_u is determined by a state transition from the joint configuration of its child nodes. Such children-to-parent transition allows to effectively model the co-occurrence of substructures among the subtrees of internal nodes [4]. The BTHMM factorizes such joint state transition as a mixture of pairwise child-to-parent transitions, to make it computationally practical. In our scenario, we model the input-conditional contribution of the l-th child to the *state-transition* of the parent node u as $P(Q_u = i|Q_{ch_l(u)} = j, x_u)$. The exact form of the input-conditional emission and transition distributions depends primarily on the input type. Here, we focus on discrete labels from a finite alphabet, which result in multinomial input-conditional distributions.

The likelihood of the Input/Output switching-parent BHTMM (IO-BHTMM) is obtained by factorizing the probability $P(\mathbf{y}^n|\mathbf{x}^n)$ using the conditional emission and transition distributions. The unknown hidden state associations $Q_u = i$ are introduced using marginalization and the final log-likelihood for the N samples writes (passages are omitted due to page limitations)

$$\mathcal{L}_c = \sum_{n=1}^{N} \sum_{i=1}^{C} \left(\sum_{u \in \mathcal{U}_n} z_u^n(i) \log P(y_u|Q_u = i, x_u) + \sum_{u' \in \mathcal{LF}_n} z_{u'}^n(i) \log P(Q_{u'} = i|x_{u'}) \right)$$
$$+ \sum_{n=1}^{N} \sum_{u \in \mathcal{I}_n} \sum_{i,j=1}^{C} \sum_{l=1}^{L} \overline{z}_{ul}^n(i,j) \left(\log P(S_u = l) + \log P(Q_u = i|Q_{ch_l} = j, x_u) \right)$$

where $P(Q_{u'} = i|x_{u'})$ is the *conditional prior distribution* for the leaves $\mathcal{LF}_n$ and

z_{ui}^n are indicator variables such that $z_{ui}^n = 1$ if $Q_u = i$ in the n-th tree and $z_{ui}^n = 0$ otherwise. Similarly, $\overline{z}_{ul}^n(i,j) = 1$ if $Q_u = i$ and its l-th child has $Q_{ch_l} = j$, while $\overline{z}_{ul}^n(i,j) = 0$ otherwise. The term $\mathcal{U}_n$ denotes the set of vertices in the n-th tree, while $P(S_u = l)$ is the switching parents distribution and measures the weight of the contribution of the l-th child to the state transition of node u.

Learning of the IO-BTHMM parameters can be achieved by applying Expectation Maximization (EM) to the log-likelihood $\mathcal{L}_c$: the E-step can be done efficiently by exploiting a modified version of the *reversed upwards-backwards* procedure for BHTMM [4]. The IO-BTHMM parameters are estimated based on the posterior of the indicator variables $\overline{z}_{ul}^n(i,j)$, that is the conditional expected value $\epsilon_{u,ch_l}^l(i,j) = E[\overline{z}_{ul}^n(i,j)|\mathcal{D}] = P(Q_u = i, Q_{ch_l(u)} = j, S_u = l|\mathbf{x},\mathbf{y})$ The posterior is estimated at the E-step by message passing on the structure of the nodes' dependency graph, using the following decomposition

$$\epsilon_u(i) = P(Q_u = i|\mathbf{x},\mathbf{y}) = \frac{P(\mathbf{x}_{1\backslash u}, \mathbf{y}_{1\backslash u}|Q_u = i)}{P(\mathbf{x}_{1\backslash u}, \mathbf{y}_{1\backslash u}|\mathbf{x}_u, \mathbf{y}_u)} P(Q_u = i|\mathbf{x}_u, \mathbf{y}_u), \qquad (1)$$

where the term $\mathbf{x}_u$ ($\mathbf{y}_u$) identifies the input (output) subtree rooted at node u, while $\mathbf{x}_{1\backslash u}$ denotes the input tree without the $\mathbf{x}_u$ subtree. Let us define $\beta_u(i) = P(Q_u = i|\mathbf{x}_u, \mathbf{y}_u)$ as the *upward probability*, that is computed during a preliminary bottom-up recursion on the tree structure. Then the posteriors can be computed with an additional downward recursions based on the $\beta_u(i)$ values [4]. Due to space limitations, we briefly discuss only the resulting update equations: further details can be obtained from the BHTMM technical report[1].

The parameters of the *upwards* recursion are recursively computed as

$$\beta_u(i) = \frac{P(y_u|Q_u = i, x_u) \sum_{l=1}^{L} P(S_u = l)\beta_{u,ch_l}(i)}{\sum_{j=1}^{C} P(y_u|Q_u = j, x_u) \sum_{l=1}^{L} P(S_u = l)\beta_{u,ch_l}(j)}, \qquad (2)$$

where $\beta_{u,ch_l}(j) = \sum_{j=1}^{C} P(Q_u = i|Q_{ch_l} = j, x_u)\beta_{ch_l}(j)$ is an auxiliary parameter. For leaf nodes the numerator in (2) reduces to $P(y_u|Q_u = i, x_u)P(Q_u = i|x_u)$. The upwards pass concludes at the root node, where we start the *downward recursion* by setting $\epsilon_1(i) = \beta_1(i)$ (follows from its definition). For each child node ch_l, going downwards, we compute the joint posterior

$$\epsilon_{u,ch_l}^l(i,j) = \frac{\epsilon_u(i)\beta_{ch_l}(j)P(S_u = l)P(Q_u = i|Q_{ch_l} = j, x_u)}{\sum_{l'=1}^{L} P(S_u = l')\beta_{u,ch_{l'}}(i)} \qquad (3)$$

and obtain the posterior $\epsilon_{ch_l}(j)$ by marginalization. Parameter update is computed at the M-step: for an IO-BHTMM model this can have an exact analytical solution depending on the form of the input/output labels. Here, we consider discrete input and output labels drawn from a finite multinomial alphabet, which ensures a closed-form solution for the M-step. The resulting update equations can be straightforwardly obtained from the posterior in (3) (see [1] for details).

[1] www.di.unipi.it/~bacciu/TRBHTMM.pdf

For instance, for a multinomial input label, the update equation for the input-conditional state-transition is

$$P(Q_u = i|Q_{ch_l} = j, x_u) = \frac{\sum_{n=1}^{N} \sum_{u \in \mathcal{U}_n} \sigma_{uk}^{n} \epsilon_{u,ch_l}^{l,n}(i,j)}{\sum_{n=1}^{N} \sum_{u \in \mathcal{U}_n} \sum_{j'=1}^{N} \sigma_{uk}^{n} \epsilon_{u,ch_l}^{l,n}(i,j')}. \tag{4}$$

where $\sigma_{uk}^{n} = 1$ if $x_u = k$ in the n-th tree and $\sigma_{uk}^{n} = 0$ otherwise.

A trained IO-BHTMM can be used to predict an output structure, given a new input tree, by means of the Viterbi recognition algorithm. By drawing on the *reversed Viterbi* in [4], we devise a recognition algorithm for IO-BHTMM that is capable of estimating the most likely joint assignment of hidden states $\mathbf{q}^*$ and output structures $\mathbf{y}^*$, given an input tree $\mathbf{x}$. In other words, we seek

$$\max_{\mathbf{y},\mathbf{q}} P(\mathbf{y},\mathbf{q}|\mathbf{x}) = \max_{i} \left\{ \delta_u(i) \max_{\substack{\mathbf{y}_{1\backslash \mathbf{CH}(u)} \\ \mathbf{q}_{1\backslash u}}} \left\{ P\left(\mathbf{y}_{1\backslash \mathbf{CH}(u)}, \mathbf{q}_{1\backslash u} | Q_u = i, \mathbf{x}_{1\backslash \mathbf{CH}(u)}\right)\right\} \right\}, \tag{5}$$

where $\mathbf{CH}(u)$ is the set of all children of node u, $\mathbf{y}_{\mathbf{CH}(u)}$ denotes the output subtrees rooted at each of the child of u, while $\mathbf{y}_{1\backslash \mathbf{CH}(u)}$ is the observed tree without the child subtrees of node u. The maximization in (5) can be efficiently computed by an upward recursion exploiting the recursive term

$$\begin{aligned}\delta_u(i) &= \max_{\mathbf{y}_{\mathbf{CH}(u)}, \mathbf{q}_{\mathbf{CH}(u)}} P(\mathbf{y}_{\mathbf{CH}(u)}, \mathbf{q}_{\mathbf{CH}(u)}, Q_u = i|\mathbf{x}_u) \\ &= \max_{\substack{j_1,\dots,j_l \\ y_1,\dots,y_l}} \left\{ \sum_{l=1}^{L} P(S_u = l) P(Q_u = i|Q_{ch_l} = j_l, x_u) \prod_{v=1}^{L} P(y_{ch_v}|j_v, x_{ch_v}) \delta_{ch_v}(j_v) \right\}.\end{aligned}$$

The basis of the recursion is at the leaves where $\delta_{u'}(i) = P(Q'_u = i|x_u)$, that is equivalent to the input-conditional prior. The $\delta_u(i)$ recursion ends at the root, where the second term in (5) evaluates to $\max_{i_1,y_1} P(y_1|Q_1 = i_1, x_1)$: an additional downward recursion allows to select the most likely hidden state and output label for the rest of the nodes. The $\delta_u(i)$ maximization can be efficiently approximated as in [4] by treating the contribution of each child in separation, i.e. allowing a parent node u to separately determine the maximum i_l and y_{ch_l} for each children ch_l, using such estimated values to compute $\delta_u(i)$.

3 Experimental Analysis

Experiments on two data sets from the 2005 and 2006 INEX Competition [5] have been performed to benchmark IO-BHTMM against the BTHMM [4] and the top-down HTMM model (THTMM) [6] on classification tasks. INEX 2005 comprises 9,361 trees, 11 classes, with 366 possible node labels. INEX 2006 comprises 12,107 trees, 18 classes, and 65 possible labels. Standard training and test sets are available for both datasets [5], with a 50%-50% split. IO-BHTMM has been trained to transform an input INEX tree into an isomorphic output tree having the corresponding class label on each of its nodes. Given

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

the discrete nature of the input/output labels, all the IO-BHTMM distribution can be modeled by multinomials. Prediction for a test tree is obtained by generating the output tree with the Viterbi. Two approaches have been tested: the former (*root*) categorizes the test tree with the label predicted by the root; the latter (*vote*) selects the most voted label among the nodes. A different BHTMM/THTMM model has been trained for each class: test predictions are obtained by selecting the model with the highest Viterbi likelihood. Table 1 shows the classification error on the test set, averaged over 5 repetitions, as a function of the hidden states' number C. Results on INEX 2005 highlights that IO-BHTMM can successfully exploit the discriminative information in the input labels, when enough hidden states are provided. Notice that different classes have almost identical distributions of the input labels: hence they are not, by themselves, discriminative. IO-BHTMM consistently reduces the error even if forced to encode the input structures within a state space smaller than the class number ($C = 10$ vs. 11/18 classes), thus achieving a notable compression of the structural information. The error reduction is statistically significative, as shown by the non-overlapping error intervals (in brackets). Conversely, BHTMM and THTMM use up to $C = 10$ states per class, as there is a different model for each class, with no clear advantage: note that when $C = 2$, IO-BHTMM has only 2 states to encode 11 classes while BTHMM and THTMM have 2×11 states. In INEX 2006, IO-BHTMM does not show a clear advantage over BHTMM, given that it is an extremely challenging benchmark with a random classifier baseline of 5.5%. Moreover, the beneficial effect of input-conditioning is mitigated by the characteristics of the INEX 2006 input labels, that are fewer and less discriminative that those of INEX 2005. In an attempt to allow more space for coding the substructures, we have increased C up to 20 states: only INEX 2005 benefitted, with the mean classification error dropping to 9.82%, while INEX 2006 maintained a 72.53% error. We have also tested a different encoding for the output tree, where a node u emits a vector of booleans such that the k-th label component is equal to 1 only if the substructure rooted in u can be found in trees of class k. This task is modeled by an emission comprising $K = 18$ independent binomials: again no clear reduction of the classification error has been shown on INEX 2006, although such an approach can be useful to shed light on discriminative substructures in the data.

4 Conclusion

We have proposed a novel input-driven generative model for structured data. The results of a preliminary experimental assessment show a good potential in capturing more discriminative structural information than its homogenous counterpart. Indeed, the proposed approach seems more effective when the input labels encode some form of discriminative information. In the proposed classification scenario, the voting mechanism seems to have a clear advantage over the root-only prediction: this is motivated by the fact that the output predicted for the substructures is often coherent with the target classification,

Hidden States	IO-BHTMM root	IO-BHTMM vote	BHTMM	THTMM
		INEX 2005		
$C = 2$	38.09 (1.24)	32.60 (2.24)	**32.20** (7.17)	34.28 (5.66)
$C = 4$	27.09 (4.86)	**19.66** (2.17)	24.98 (5.89)	23.40 (4.89)
$C = 6$	16.45 (2.84)	**15.10** (2.77)	22.91 (3.64)	30.50 (9.33)
$C = 8$	16.43 (3.88)	**13.31** (2.75)	18.11 (3.02)	27.36 (6.53)
$C = 10$	12.18 (3.57)	**11.43** (2.93)	18.93 (3.18)	28.92 (4.53)
		INEX 2006		
$C = 2$	83.05 (4.34)	76.28 (0.99)	**73.81** (0.67)	76.70 (2.72)
$C = 4$	76.60 (0.63)	**75.01** (0.83)	76.50 (2.79)	77.87 (2.91)
$C = 6$	75.18 (1.74)	74.94 (0.95)	**74.25** (2.99)	76.73 (1.93)
$C = 8$	74.58 (1.17)	**73.90** (0.47)	74.02 (0.79)	77.18 (2.66)
$C = 10$	73.23 (1.26)	**72.77** (0.88)	73.06 (2.78)	77.90 (1.83)

Table 1: Results on the INEX 2005 and 2006 datasets: average classification error on the test set (best in bold); standard deviation is between brackets.

even when the root node emits the incorrect output label. This is the result of the contextualized predictions generated by the Viterbi, where the outputs predicted in a substructure are interpreted on the basis of the label generated for their parents. Hence, non-discriminative substructures recurring over different classes tend to be appropriately contextualized, yielding to correct predictions. This marks a notable advantage over the homogenous BHTMM and THTMM, where each competing model is trained only on positive examples from a target class. As a result, the homogenous models tend to get confused by non-discriminative substructures, yielding to high likelihoods also for trees outside their target class. Overall, the experimental results suggest that input-driven generative models have a good potential when applied to tree-structured data. Based on this results, we intend to tackle the long-standing open problem of learning non-isomorph input-output transformations for tree-structured data [7].

References

[1] Y. Bengio and P. Frasconi. Input-output hmms for sequence processing. *IEEE Trans. Neural Networks*, 7(5):1231–1249, 1996.

[2] Y. Li and H.Y. Shum. Learning dynamic audio-visual mapping with input-output hidden markov models. *Multimedia, IEEE Transactions on*, 8(3):542–549, 2006.

[3] Y. Bengio. Markovian models for sequential data. *Neural Comp. Surv.*, 2:129–162, 1999.

[4] D. Bacciu, A. Micheli, and A. Sperduti. Bottom-up generative modeling of tree-structured data. *Neural Information Processing. Theory and Algorithms*, pages 660–668, 2010.

[5] L. Denoyer and P. Gallinari. Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents. *SIGIR Forum*, 41(1):79–90, 2007.

[6] J.B. Durand, P. Goncalves, Y. Guedon, I. Rhone-Alpes, and F. Montbonnot. Computational methods for hidden Markov tree models-an application to wavelet trees. *IEEE Trans. Signal Process.*, 52(9):2551–2560, 2004.

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