Graphical Models for Discrete Hidden Markov Models in Speech Recognition

Antonio Miguel, Alfonso Ortega, Luis Buera and Eduardo Lleida

Communications Technology Group (GTC)
Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
{amiguel, ortega, lbuera, lleida}@unizar.es

Abstract

Emission probability distributions in speech recognition have been traditionally associated to continuous random variables. The most successful models have been the mixtures of Gaussians in the states of the hidden Markov models to generate/capture observations. In this work we show how graphical models can be used to extract the joint information of more than two features. This is possible if we previously quantize the speech features to a small number of levels and model them as discrete random variables. In this paper it is shown a method to estimate a graphical model with a bounded number of dependencies, which is a subset of the directed acyclic graph based model framework, Bayesian networks. Some experimental results have been obtained with mixtures of graphical models compared to baseline systems using mixtures of Gaussians with full and diagonal covariance matrices.

Index Terms: Graphical Models, Bayesian networks, Maximum Likelihood, Expectation maximization.

1. Introduction

In this paper we propose the use of the graphical model approach to discover underlying dependency structures in the observation generation process of the HMM states. Usually, the multivariate distribution of the random feature vectors in speech recognition is assumed to follow a mixture of Gaussians with diagonal covariance matrix. When more accuracy is required, and enough training data are available, full covariance matrices are considered, which take into account pairs of variables. We propose to use a model which can take into account larger order dependencies between the components of the feature vector by using discrete variables and graphical models.

Most of today’s automatic speech recognition systems are based on statistical approaches for modeling the process of emission of observations in the HMM. From the statistical point of view the speech signal can be considered a very complex process due to its non-stationary, the variability across speakers, environments, or even for a same individual. The exact modeling of all of these sources of uncertainty in a brute force approach is not affordable. We need to provide mechanisms able of making approximations and generalize. In the field of graphical models there are many interesting solutions to deal with large number of variables and all their complex interactions [1]. Nowadays, it is a very active field of research with many applications in pattern recognition, machine learning and intelligent systems. Many authors have contributed to this line of research from different areas, but the application of these powerful tools in acoustic modeling is limited. Graphical models have been applied by [2] as an alternative to the HMM inter-frame independence assumption and to model the covariance matrix internal dependency structure.

In this work we want to study probabilistic models in which variables may depend on more than one variable. This can be done is discrete observations are assumed as output of the states of the HMMs as [3]. Nevertheless, we propose to capture the dependencies between the discrete variables by using the graphical model approach. The objective is to learn approximations to the joint distributions of the speech features, since the exact joint distribution is not feasible. Depending on the number of dependencies that we keep, we can obtain models with different levels of complexity. It can shown that for a certain fixed level of complexity, the best approximation from the maximum likelihood point of view is given by a factorization described by a directed acyclic graph or Bayesian network with a constrained number of parents per node. We will focus on a subset of the Bayesian networks in which there is a maximum limit in the number of parents per node, called bounded treewidth Bayesian networks (BTBNs) [4].

This paper is organized as follows. In Section 2 discrete variable distributions and factorized functions for speech recognition are presented. Section 3 explains constrained order dependency models. Section 4 shows the estimation procedure for bounded treewidth Bayesian networks and an approximated algorithm to build the graph. Section 5 presents the factorized distributions model embedded in a hidden variable structure like a finite mixture. In Section 6 presents an experimental study to show the accuracy of the models in speech recognition. Conclusions and future lines are in Section 7.

2. Discrete Models

Discrete variables have been previously used in speech recognition. In [3] it was shown a method to obtain fast and accurate models based on discrete random variables, which where obtained from a set of vector quantifiers. The model used for the states of the HMM was a mixture of discrete distributions in which the variables were considered independent given the component of the mixture. As we will see, there are some differences with this work in the complexity of the model and the quantization process.

In order to perform the quantization process in this work, a simple process is proposed. First we take the complete training corpus and estimate the histogram of each feature. Then, we find a number of areas with approximately similar probability mass. The limits between these areas will serve to build the quantizer. This process can be seen equivalent to construct a histogram equalization transformation function with a uniform target distribution followed by a uniform quantization. The ob-
jective is to build a quantizer so that each quantized level rep-
resents the same amount of mass of probability from the input
signal.

2.1. Factorized Probability Density Functions

The probabilistic model that we want to examine is the discrete probability density function (pdf) in the states of a HMM. Also it can be associated to a component in a mixture for each state as it is shown in Section 5.

The pdf to model is the corresponding to the joint distribution, \( p(x) \), of a \( D \) dimensional feature vector \( x = (x_1, x_2, \ldots, x_D) \), where each component of the vector \( x_d \) is a discrete variable with outcomes \( x_d \in \{1, \ldots, M\} \), with \( M \) the number of levels after the quantization.

In the case of discrete random variables, it is possible to define the exact joint pdf, but the parameter size grows exponentially with the dimension. In [5], it was proposed a method for storing pdfs based on a convenient “factorization” of the exact pdf. In [5] the dependencies where limited to a maximum of one. In this work we will investigate pdfs with a maximum number of dependencies of two.

2.2. Graphical models

The objective in the graphical model approach is to find a way of describing the interactions and independences between the variables of a probabilistic model to capture useful information.

The DAGs (Directed Acyclic Graphs) can be used to describe the most important information and dependencies between variables in a factorization. A probabilistic model described by a DAG is a Bayesian networks (BNs), [1], which can represent knowledge in a powerful way, especially when dealing with a large number of sources of information and all their complex interactions.

To build a factorization model, each variable in the model \( x_d \) is associated to a node in the graph \( v \). The graph is defined as the pair \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \), where \( \mathcal{V} \) is the set of vertices and \( \mathcal{E} \) the set of edges. Therefore the size of the graph is \( |\mathcal{V}| = \sum_d v_d = D \), the dimension of the vectors. A directed acyclic graph \( \mathcal{G} \) associated to a probabilistic model \( p(x) \) defines a unique factorization approximation for the pdf in the following way

\[
p(x) \approx \prod_{v=1}^V p(x_v|\pi(v)) = \prod_{v=1}^V \prod_{\pi(v)} p(x_v|\pi_1(v), \ldots, \pi_v(v)), \tag{1}
\]

where the set of dependencies associated to node \( v, \pi(v) \), are the parents of the node \( v \) in the graph of components \( \{\pi_1(v), \pi_2(v), \ldots, \pi_v(v)\} \).

3. Bounded Treewidth Bayesian Network Models

The number of parent nodes of a node defines the dependencies of the associated variable with respect to other variables and it is defined in the graph. The kind of model that we propose to use for speech recognition is a subset of the Bayesian networks, the bounded treewidth Bayesian networks (BTBN) [4].

The complexity of the target factorization can be controlled as a parameter the order \( r \), which defines the BTBN(\( r \)) model. The associated graph in a BTBN is a DAG with a limited number of \( r \) parents per node, \( |\pi(v)| \leq r \).

For orders smaller than \( r = 2 \), there have been many previous uses in pattern and speech recognition. If there are not dependencies in the model, then \( \pi(v) = \emptyset \) for all the variables and \( |\pi(v)| = 0 \), then the model is called naive Bayes and it is often used in mixture models. The case where \( |\pi(v)| \leq 1 \), is known as Chow-Liu tree [5] and also has been used in mixture models [6]. In this work we explore models of greater complexity in the number of dependencies. For simplicity in the notation we are going to express the model probability and estimate the parameters for an order \( r = 2 \), but the method is also applicable to larger orders.

We propose the adjacency matrix of the graph, \( A \), to establish a more convenient notation. The term \( p(x_v|\pi_1(v), \pi_2(v)) \) for \( r = 2 \) in (1) can be expressed as

\[
p(x_v|\pi_1(v), \pi_2(v)) = \prod_{v'} \prod_{v''} \prod_{\pi(v')} p(x_v|x_{v'}, x_{v''})^{a_{v', v''}}, \tag{2}
\]

where \( a_{v', v''} \) is a component of adjacency matrix which is equal to one if the node \( v' \) is a parent of node \( v, a_{v', v''} = 1 \iff x_{v'} \in \pi(v) \) and it is equal to zero otherwise. For simplicity, we reduce the notation in the upper limits of the products and sums.

We assume multinomial distributions for the terms (2) as

\[
p(x_v|\pi_1(v), \pi_2(v)) = \prod_{v'} \prod_{v''} \prod_{\pi(v')} p(x_v|x_{v'}, x_{v''})^{a_{v', v''}}(\delta_{x_{v'}, x_{v''}}), \tag{3}
\]

In the previous expression, we assume that the random variable \( x_{v'} | x_{v''} = m', x_{v''} = m'' \) follows a multinomial distribution of prototype vector \( p_{v', v''} | m', m'' \), whose components are the probabilities

\[
p_{v', v''} | m', m'' = P(x_v = m | x_{v'} = m', x_{v''} = m''), \tag{4}
\]

with the constraint \( \sum_{m} P_{v', v''} | m', m'' = 1 \). The set of all the multinomial prototypes is called \( \mathcal{P} \).

In [7], we express the log likelihood function, \( L(\Theta; X) \), for the parameters \( \Theta \), and a training set, \( X = \{x_1, \ldots, x_n\} \), where the parameters are \( \Theta = (P, A) \). Therefore, we can see that the model is completely defined once we know the multinomial prototypes, \( P \), and the adjacency matrix, \( A \).

4. Parameter Estimation

In order to estimate the optimum set of parameters to maximize the log likelihood function we have to solve the following optimization

\[
\{\hat{P}, \hat{A}\} = \arg \max_P L(P, A; X), \tag{5}
\]

subject to \( \sum_{m=1}^{M} P_{v', v''} | m', m'' = 1 \) for all \( v', v'' = 1, \ldots, V \) and \( m', m'' = 1, \ldots, M \), and subject to \( A \in \text{BTBN}(2) \).

Obtaining for the Multinomial parameters the following estimation

\[
\hat{P}_{v', v''} | m', m'' = \frac{\sum_{n} \delta_{x_{n,v'}, m'} \delta_{x_{n,v''}, m''}}{\sum_{n} \delta_{x_{n,v'}, m'} \delta_{x_{n,v''}, m''}}, \tag{6}
\]

which can be interpreted as the probability \( \hat{P}(x_v = m | x_{v'} = m', x_{v''} = m'') \). The previous solution shows that the optimum parameter subset \( \hat{P} \) is obtained independently of the topology of the graph.

4.1. Graph structure

The optimum set of parameters \( \hat{A} \), provides the edge set so that the graph can will be fully characterized. From (5) we can obtain the following expression [7]

\[
\hat{A} = \arg \max_A \sum_{v', v''} (a_{v', v''} | a_{v', v''}) \cdot \hat{I}(x_v|x_{v'}, x_{v''}), \tag{7}
\]
where the BTBN(2) topology constraints have been used and

$$I(x_v|x_{v'}, x_{v''}) = \sum_{x_{v'}, x_{v''}} \hat{p}(x_v, x_{v'}, x_{v''}) \log \frac{\hat{p}(x_v, x_{v'}, x_{v''})}{\hat{p}(x_v)}$$

where the previous expression is the Kullback-Leibler (KL) divergence between two distributions: $p(\cdot, \cdot, \cdot) = p(x_v, x_{v'}, x_{v''})$, the exact distribution, and $\hat{p}(\cdot, \cdot, \cdot) = p(x_v) \cdot p(x_{v'}, x_{v''})$, an independence approximation. It also can be seen as the mutual information between the random variables $(x_v)$ and $(x_{v'}, x_{v''})$, this last variable can be considered as a joint process. If this value is close to zero, this means that the approximation taken from $p(\cdot)$ to $p(\cdot)$ is close to be true, and there is not a loss of information in doing so. If the value is high, this will mean that the variables in the $p(\cdot)$ distribution should be considered jointly since they provide useful information.

It is interesting that as in [5] we can obtain the same solution (7) by minimizing the KL divergence between the approximate and the exact model $D(p(x)||\hat{p}(x))$ as shown in [7].

4.2. Approximated algorithm for graph building

It has been shown previously that the estimation of Bayesian networks is NP-complete [8]. Therefore, the optimization expression (7) has not a closed form solution, but a fast approximate algorithm to estimate the best graph is proposed in this paper. The objective is to find an algorithm to obtain the best graph in terms of maximum likelihood, which is equivalent to find the set of edges $\hat{A}$ that maximize expression (7). This problem is trivial for order $r = 0$, (the naive Bayes model), and can be solved exactly in polynomial time for order $r = 1$, where we will obtain a special kind of graph, a tree, which is shown in [5]. For higher orders such as $r = 2$, the problem becomes intractable [8] and only approximate algorithms can be applied for large problem sizes $V$.

Algorithm 1 Approximate optimum graph for a BTBN(2)

Input: Random samples $X$

Output: The graph $\hat{A}$ of the type BTBN(2)

1. Initialization

Estimate distributions $\hat{p}(x_v, x_{v'}, x_{v''})$ and $I(x_v||x_{v'}, x_{v''})$

Graph matrix: $A = 0$

Set of non assigned nodes, $\mathcal{N} = \mathcal{V}$

Order $I(x_v||x_{v'}, x_{v''})$ values in descending order

2. Search edges

while $|\mathcal{N}| > 1$
do

Get next ordered triple $(v, v', v'')$

if $v \in \mathcal{N}$ then

$A' = A$

Add edges $(v', v)$ and $(v'', v)$ to $A'$:

$a'_{v', v} = 1$, $a'_{v'', v} = 1$

if $|I - A'| \neq 0$ then

$a_{v', v} = 1$, $a_{v'', v} = 1$

$\mathcal{N} = \mathcal{N} \setminus v$

end

end

The approximate Algorithm 1, can be explained as follows. First the joint distributions $\hat{p}(x_v, x_{v'}, x_{v''})$ and the KL divergences $I(x_v||x_{v'}, x_{v''})$ have to be calculated in a initializing phase. Then, the values of the KL divergence or mutual information terms are ordered, so that we start the algorithm trying the higher triples. The next step is the approximate search of the edges to construct the matrix $\hat{A}$ trying to optimize (7). This means that we need to have a maximum value of the sum of partial KL divergences for all the edges in the graph, while keeping the graph acyclic. This is done in a loop by adding consecutively pairs of edges $(v', v)$ and $(v'', v)$ following the previous descending order. Before adding them to the solution, it is checked that the addition of both pairs does not form a cycle using the determinant $|I - A'|$, which is equal to zero if there is a cycle, being $I$ the identity matrix of size $V$.

There exists an approach [9] to discover conditional independences and the dependency graph in data sets. In order to do so, there is a first step involving the computation of terms $I(x_v||x_{v'})$, as in [5], as a first approximation. In later passes of that algorithm CI (conditional independence tests) are performed to overcome the first initial approximation.

5. Mixtures of CDAGs

Similarly to [3], we define now the mixture of discrete pdfs, which can be seen as a mechanism to augment the ability of basic models to cope with complex modes of variation in the data. A mixture of $C$ BTBN(r) components can be defined as

$$p(x) = \sum_{c=1}^{C} w_c \cdot p(x|A_c, P_c)$$

where $A_c$ and $P_c$ are the graph and the multinomial prototypes associated to the component $c$.

We use the EM algorithm to estimate the parameters of a mixture of constrained models. Then as shown in [7], the E step is done computing the expected value for the hidden variable, $Z$, component labels given the training data

$$\langle z_{x, c}^{(k)} \rangle_{x, c}^{(k)} = \frac{w_c^{(k)} \cdot p(x|Z_{x, c}^{(k)} = c, A_c^{(k)}, P_c^{(k)})}{\sum_{c'} w_{c'}^{(k)} \cdot p(x|Z_{x, c'}^{(k)} = c', A_{c'}^{(k)}, P_{c'}^{(k)})}$$

Given the expected value for the component labels, the estimation update for the mixture weights is the same as in the Gaussian case. The M step to estimate the next value of the multinomial prototypes is

$$P_{x, v', v'', m, m', m''}^{(k+1)} = \sum_n \langle z_{x, c}^{(k)} \rangle_{x, c}^{(k)} \cdot \langle z_{x, m, m', m''}^{(k)} \rangle_{x, m, m', m''}^{(k)} \cdot \frac{\delta_{x, m} \delta_{x, m'} \delta_{x, m''}}{\delta_{x, m} \delta_{x, m'} \delta_{x, m''}}$$

Finally, the graph is computed following the optimization expression

$$A_c^{(k+1)} = \arg \max_{A} \sum_{x, v', v''} \langle a_{v', v} \cdot a_{v'', v} \rangle I(x_v||x_{v'}, x_{v''})$$

6. Experiments

The proposal in this paper has been evaluated on the Aurora3 Spanish database [10]. The experiments are done with the advanced ETSI front-end [11]. The baseline system was based on HMM word models of 16 states and 3 components per mixture for the digits and 1 and 3 states with 6 components per mixture for the inter-word and begin-end silence models respectively. The Algorithm 1 and (11) were used to estimate the parameters. The equalization-quantization process, was performed
based only on the training data and was fixed during test. In order to provide additional robustness, we used smoothing [12]. We also have applied independence between dynamic streams given the component in the mixture.

We compare different mixtures of BTBN approaches with two baseline systems based on mixtures of diagonal or full covariance Gaussians. The results are displayed on Figure 1, where we can observe the behavior of the discrete variable approaches with different number of quantization levels $M$ and orders $r$. In these experiments, the discrete models perform better than both baseline systems for many configurations. We can observe that the BTBN(2) model performs well in the well matched (WM) and the medium mismatch (MM) cases, but in the high mismatch (HM) case, there is not a big difference given the order $r$. This can be explained since we could only expect good results in matched conditions, when the mutual information calculations that we perform in the estimation process are consistent in training and test. Then, in matched conditions we can get improvements by increasing the order of the BTBN, similarly to what is done when upgrading the diagonal to the full covariance matrix in Gaussian models.

Nevertheless, we can observe other interesting point. For a relatively small number of levels, $M = 3$, the systems perform well in HM conditions, while in the same conditions the full covariance Gaussian suffers a great degradation. This result is more related to the noise robustness achieved when a feature is quantized with only three levels. In the BTBN approach we have a new mechanism to control the sensitivity to mismatch which is the quantization process.

We also can see that when the number of levels grows the BTBN(2) system degrades faster. The number of examples available for training can explain this fact. For higher order dependencies it is more difficult to estimate probabilities when the number of levels grows, since the parameter space grows exponentially with the order $r$. Since the number of examples for training in Aurora3 is constant, we will have less examples to estimate the probabilities.

7. Conclusion

In this paper it has been shown a method to model high dimensional discrete distributions which is based on the assumption of a factorization model where the number of dependencies is limited. The generalization ability of these factorizations had been previously shown in previous works [4]. We have adapted a previous maximum likelihood solution for a constrained order of one, [5], to larger orders, where the estimation process has to be approximated. The result has been a very interesting class of model for discrete feature vectors with a good accuracy and benefits such us a low transmission rate for the features, which we will continue to enhance in future works oriented to DSR systems. The behavior in noise conditions has also been found to be good in matched and unmatched training conditions, outperforming the baseline in many cases. This effect is more noticeable with low number of levels, since quantized values are less sensitive to the noise uncertainty than continuous values.

8. References