Stochastic Methods and Algorithms for Analysis in Communication Process

MIHAELA ADELA ŢĂRANU, GEORGE LĂZĂROIU, SOFIA BRATU
Spiru Haret University
Italiana No. 28, Bucharest
ROMÂNIA
adelataranu@gmail.com, phd_lazaroiu@yahoo.com, sofia_pub@yahoo.com
www.spiruharet.ro

Abstract: In the present study we addressed the signal processing communication. Spontaneous communication recognition and understanding is a difficult problem and it must be considered not only the information signal itself but also the context in which it appears. Given the context, people can understand the communication signal even when it is affected by noise. Understanding the context of communication based on an extensive knowledge of the world and this was the source of the difficulty in 40 years of research. Communication recognition by computer is currently possible only for individual words, by limiting the vocabulary or the number of speakers or restricting how to form sentences.

Key-Words: Stochastic Methods, Communication Methods, Analysis and Processing Signals

1 Introduction

In the first part we present the problem of communication recognition and we describe a powerful statistical method used successfully by most systems automatic communication recognition. We refer to the Hidden Markov model (HMM), which is an extension of Markov model in that incorporates a double stochastic process with a process associated states (which is not directly observable) and a process that produces the sequence of observations.

2 Problem Formulations and Solutions

In communication recognition problems, observations are in vectors as a continuous space, each state of the model has an associated density function and the function of the first candidate discrete composition of Gaussian densities, because density can approximate any function.

The three basic problems of Hidden Markov model known as the problem assessment, problem decoding and learning problem.

If we want to use HMM for recognition of forms, then the sequence of observations \( \beta = \beta_1 \beta_2 \ldots \beta_t \) is given a sequence of observations \( \beta = \beta_1 \beta_2 \ldots \beta_n \), and in this way is reduced the maximization \( \pi ( \xi, \beta, F) \), which is equivalent to maximizing \( \pi ( \xi, \beta, F) \). This problem is very similar to the problem in an optimal way graph resolving dynamic programming method. As a consequence of this observation, for determining the best sequence of states for an HMM can be used an algorithm based on dynamic programming, known as the Viterbi algorithm and decoding algorithm \( \alpha \).

To determine the most likely sequence of states \( \xi = \xi_1 \xi_2 \ldots \xi_t \), is given a sequence of observations \( \beta = \beta_1 \beta_2 \ldots \beta_n \), and in this way is reduced the maximization \( \pi ( \xi, \beta, F) \), which is equivalent to maximizing \( \pi ( \xi, \beta, F) \). This problem is very similar to the problem in an optimal way graph resolving dynamic programming method. As a consequence of this observation, for determining the best sequence of states for an HMM can be used an algorithm based on dynamic programming, known as the Viterbi algorithm. In practice, this HMM algorithm is applied for evaluation and, in most cases, to obtain the same solution forward algorithm. However, instead of summing the probabilities of states that have subsequence the same purpose (as the forward algorithm), Viterbi algorithm chooses and saves the best afterwards. In building
communication recognition systems, we have an essential role of language stochastic models. To improve the recognition accuracy, we need to calculate the probability of occurrence of a sequence of words \( \omega = \omega_1 \omega_2 \ldots \omega_n \). The main purpose of stochastic language models is that each sequence of words to be associated with a likely reflecting the frequency of sequences in that language.

\( \pi(\omega) \) can be decomposed as follows:

\[
\pi(\omega) = \pi(\omega_1, \omega_2, \ldots, \omega_n) = \pi(\omega_1 \mid \omega_0, \omega_2) \pi(\omega_1) \ldots \pi(\omega_n \mid \omega_1, \omega_2, \ldots, \omega_{n-1}) = \prod_{i=1}^{n} \pi(\omega_i \mid \omega_{i-1})
\]

where \( \pi(\omega_1 \mid \omega_0, \omega_2, \ldots, \omega_{n-1}) \) is likely to follow \( \omega_1 \), given the sequence \( \omega_0, \omega_2, \ldots, \omega_{n-1} \).

For a vocabulary of \( V \) words, estimated \( V \) and values to determine the full \( \pi(\omega_1 \mid \omega_0, \omega_2, \ldots, \omega_{n-1}) \). In reality, these probabilities are impossible to quantify, because most of sequence \( \omega_0, \omega_2, \ldots, \omega_{n-1} \) are unique or occur very infrequently. A practical solution is to suppose that \( \pi(\omega_n \mid \omega_0, \omega_2, \ldots, \omega_{n-1}) \) depends on a number of previous word \( \omega_{n-1}, \omega_{n-2}, \ldots, \omega_1 \). Thus, it appears most used language model called stochastic n-gram. If occurrence of a word depends on the two previous words, we have a trigram language model:

\( P(\omega_n \mid \omega_{n-1}, \omega_{n-2}) \).

Similarly, we model unigram \( \pi(\omega_1) \) or bigram \( \pi(\omega_n \mid \omega_{n-1}) \). In practice, model trigram is most often used with good results, since most words have a strong dependence of the two preceding words.

In practical cases, for a large vocabulary follows a pattern that has many states, making Viterbi algorithm slow. It is possible to reduce state-space without compromising results. This is achieved by a beam search, ie hypotheses with a probability below a certain level are ignored. Decoding algorithm \( \alpha \) type implements a best-first search in a tree or a lattice defining a default sequence of words accepted by a language. In communication recognition, acoustic signal having \( \beta \), want to find the most likely sequence of words \( \omega \) that have produced \( \beta \):

\[
\omega = \arg \max_{\omega} \pi(\beta \mid \omega) \pi(\omega).
\]

Since we have \( \omega = \omega_1, \omega_2, \ldots, \omega_n \), we can see the problem as a search for a path into a tree of whose edges are labeled with words from vocabulary \( \mathcal{V} = \{ \omega_1, \omega_2, \ldots, \omega \mid \mathcal{V} \} \).

Algorithm \( \alpha \) is suitable for communication recognition problems, if finding good heuristic functions to guide search in a positive direction without exploring more nodes. Unlike the Viterbi algorithm, \( \alpha \) algorithm is not time synchronous, ie extending roads of different lengths.

Since the \( \alpha \) algorithm is not time synchronous, we need a mechanism to determine when to finish the evaluation of a word / phoneme and get to the next word / phoneme. However, we need a heuristic function to estimate input remained unprocessed. But even if we have an effective heuristic function and a mechanism for detecting the end of a phoneme/word, algorithm \( \alpha \), applied communication recognition create problems for a large vocabulary (over 60,000 words) is slow. Asynchronous time looking at expensive step is to extend the best way, because it involves calculating probabilities the occurrence of all words \( v \) of the vocabulary. The words that would reduce space may start at some time \( t \) in the acoustic signal, and this is done by mechanism that reduces the number of fast match words / phonemes candidate to expand.

Even if we have a heuristic function \( h(\cdot) \) very good to guide the search algorithm \( \alpha \), is more effective to compare the roads that have the same length as in this case comparison is based on forward accurate assessment of the acoustic signal processed by time. Thus, it was proposed a variant of \( \alpha \) algorithm working with many stacks. Each stack is time synchronous, which is maintaining for each moment of time \( t \) the signal so that it does not need comparing different lengths of roads. At time \( t \), the algorithm extracted from the stack corresponding best way, it expands by one word, and new roads to enter the stack allocated to those lengths. When the algorithm terminates, the road saved on top of last stack is the optimal pathway.

A third important issue of HMM (learning problem) is how to estimate parameters model \( F \) of input data in order to describe the best sequences of observations. It is the most difficult of the three problems, because there is no analytical method to solve the equation

\[
\Phi^* = \arg \max_{\Phi} \pi(\beta \mid \Phi)
\]

The problem can be solved instead of iterative Baum-Welch algorithm, also known as called forward-backward algorithm. To demonstrate the convergence algorithm optimal solution, we use EM algorithm (expectation maximization) which can be seen as a generalization of the method MLE (maximum likelihood estimation) when observed data are incomplete. Suppose we observe the time \( y \). To determine the parameter \( \Phi \) which maximizes \( \pi \) (\( \text{Y} = \gamma \Phi \)) we need to know every hidden \( \beta \) (where \( \beta \) is the HMM hidden sequence of states). Without knowing this information we can use estimated maximum plausibility to estimate \( F \) that maximizes
π (Y= γ Φ). To overcome this problem, we fix a value for the parameter Φ and we estimate the probability of occurrence of each β when given γ. Thus, we claim that we have complete observations (β, γ) with probability π (β = β, Y = γ Φ) to calculate Φ*, estimated maximum plausibility of Φ. Replaced Φ with Φ* and repeat the process to improve estimation.

In the learning problem we presented a technique to improve discrimination between HMM states; this is accomplished by determining the transformation linear between a collections of features in a large-scale space by a space size model.

An alternative for HMM models used to calculating probabilities represent phonemes feed-forward neural networks MLP (multilayer Perceptron). This model is a hybrid HMM-MLP, and use elements of the HMM (as a graph representation of states of pronunciation of words). But the probabilities are calculated by MLP comments. HM models presented are the basic technique used in most existing communication recognition systems.

Regarding the search algorithms presented (Viterbi and α), they have been successfully applied in various systems of recognition. Viterbi beam search was first used in 1976 in communication recognition system HARPY, but has become with the advent in 1985 of BYBLOS system. It then occurred more effectively improve time-synchronous Viterbi beam search, so that today there real-time systems for continuous communication recognition in a large vocabulary that can be implemented on personal computers.

The algorithm α was implemented first time in a communication recognition system for IBM 1983 and is successfully used in systems produced by IBM. In practice, the comparison roads different lengths, and road expansion is much more complex than described in this part. Due to the complexity, many more systems based on Viterbi beam search.

But with the advent of multi search system, this version of the algorithm α becoming very similar with time-synchronous Viterbi beam search.

General problem of automatic decoding of communication uttered by anyone in any environment is far from being resolved. In recent years, however, automatic recognition technology has reached a certain maturity and is viable in some limited areas (human-computer interaction, control of equipment, telephone, and dictation).

The second part of the thesis we applied predictive methods for modeling and recognition Communication signals. Since communication signals are nonlinear prediction model used was NARMA (Nonlinear ARMA) whose general form is given by equation

\[ y(k) = \Phi \left( \sum_{i=1}^{n} a_i y(k-i) + \sum_{j=1}^{m} \theta_j \varepsilon(k-j) \right) + e(k), \]

where y(k) is output, e(k) input, and Φ (·) is a differentiable nonlinear function. Appropriate predictors is given by

\[ \hat{y}(k) = \Phi \left( \sum_{i=1}^{n} \alpha_i y(k-i) + \sum_{j=1}^{m} \theta_j \varepsilon(k-j) \right), \]

residues being

\[ \varepsilon(k-j) = y(k-j) - \hat{y}(k-j), j = 1, 2, \ldots, \eta. \]

To determine the model parameters NARMA, we implemented three adaptive algorithms learning with neural networks. A neural network can implement a process Narma appellant is Perceptron. It consists of three layers: input, processing and output. Components input vector are weighted and summed to produce the internal activation of Perceptron, which is then passed through the nonlinear activation function Φ (usually the logistics function) to get out of the network. Perceptron training algorithm RTRL (real time recurrent learning) appellant minimizes instantaneous square error. Another algorithm, ERLS (extended recursive least squares), is based on the idea of extended Kalman filter, which can be used for training Perceptron appellant, considering the weights as a nonlinear dynamical system with state stationary.

Some interesting problem that we tried to solve was the choice of number optimal parameters of the model prediction. If the number of parameters is too small, the model fails to learn the signal. If the number is too high, then occurrences of super-learning, which the model loses its ability to generalize. There are several criteria which seeks to prevent the phenomenon of super-adjustment signal model is represented by (Brockwell and Davis, 1987).

Starting from an approach which we found in (Mandic and Chambers, 2001), we obtained a series of results on stability and convergence of learning algorithms Perceptron appellant. Since analysis of the dynamics of an adaptive system may be reduced to discovery of attractor (stable equilibrium), it is important to study the stability and convergence to such a fixed point. The neural associative memories, equilibrium locally stable (attractors) neural form memory. In this case, neural dynamics can be seen two perspectives, convergence of state variables (information retrieval) and the number, position, Local stability and domain of attraction of steady state (memory capacity).

Dynamics and convergence of learning algorithms can be explained and analyzed using the theory of fixed point. The results show that the mean squared
error convergence is required stronger condition than convergence in average weights errors. At the end of the second part we presented a series of results obtained by testing algorithms presented using send you database, which contains signals uttered by 630 Speakers in 8 dialects of English, each accounted for 10 sentences each. We noticed that predictors manage to learn the characteristics of female communication / male spoke of signal such a signal can decide whether it is a woman or a man.

The equations describing the Kalman filter, we calculated the position of plausibility for each signal and thus we could apply the Akaike criterion to determine the number of NARMA model parameters. For some signals, the model provided by the Akaike criterion had a high capacity for generalization, that good prediction signal which was not training. But this was not true for all signals. We think that in order to determine NARMA model parameters should be applied Akaike criterion subject to a maximum the mean squared error obtained while implementing involvement Perceptron model respectively.

The problem which we stopped is BSS (Blind Source Separation) and is to separate multiple signals mixed linearly independent, a series of observations, not to know how they have been mixed. An example of the acoustic field is known as the cocktail party, in that requires the separation of signals captured from multiple microphones at different speaking people. Statistical methods have been proposed for the processing of signals, but also techniques based on elements of information theory. The function Kullback-Leibler show the divergence between joint density and product densities marginal parts mixed signals, mutual entropy and Renyi entropy maximum are successfully used information criteria for signal separation algorithms independent. Also, we have taken from idea of using adaptive functions minimizing the Kullback-Leibler divergence, but detailed explanation of the algorithm is mine. This method avoids direct estimation of output signals and adjusts distributions functions activation required.

Another criterion for separation BSS problem is to minimize non-parametric estimator for mutual Renyi entropy. Signal separation process is of two phases: decorrelated signals, followed by a series of givens rotations to obtain independent outputs. Assuming that the observed signals \( z \) are processes of average 0, covariance matrix will be \( \Sigma = E [zz^T] \). If \( \Lambda \) is diagonal matrix of eigenvalues and \( \Phi \) is orthogonal matrix of their corresponding vectors of covariance matrix \( \Sigma \), then \( \omega = \Lambda^{1/2} \Phi^T \) is the de-correlation matrix for \( z \). The adaptive part is a rotation matrix which produces output \( \gamma = \rho(\theta)\beta \), where \( \beta = \omega z \) is the de-correlated vector. Output vector \( y \) obtained is two by independent components. It was shown that when matrices unmixed are mixing and rotation matrix (as happens after decorrelarea signals), independent components grouped in pairs are equivalent to independence mutual.

The algorithm that we proposed is used as a measure of independence of two random variables, square contingency test of independence \( \lambda^2 \). To use this measure and to solve the BSS problem, the process of separating the signals will consists of two stages, as in the case of Renyi entropy minimization: decorrelation of signals, followed in the second phase of givens rotations to obtain independent output signals. Because square contingency is calculated at each step number and not by an expression that can compared with matrix derived through un-mixed, minimization is done by any non-gradient optimization technique (eg golden section method).

If the maximum entropy method, the output vector \( \gamma \) is transformed into a vector \( \zeta \) with a function nonlinear, monotone, bounded invertible and \( \phi (-) \). This ensures the Shannon entropy \( h(\zeta) \) for any un-mixed matrix \( \omega \). For a function \( \phi(-) \) is chosen convenient method for maximum entropy source \( \xi \), estimated through maximizing the entropy \( h(\zeta) \) in comparison with \( \omega \).

## 3 Conclusion

In the past years was made good approximation proposed solutions based on different assumptions the original signals were obtained and efficient adaptive algorithms starting from different positions cost, such as Kullback-Leibler divergence, Renyi mutual entropy, maximum entropy, etc. The algorithm that we proposed, based on minimizing the test square contingency \( \lambda^2 \), it had comparable results with other algorithms that we described in this part.

But the standard formulation of ICA (algorithm which using a geodesic approach with Hessian and Jacobian models) has several limitations: the number of recorded signals (sensors) be less than or equal to the number of source signals, there is additive noise in sensors; sources are modeled as independent random variables. It is clear that the problems real this is not credible and, although the standard ICA algorithms give good solutions approximation, it is necessary to develop new algorithms to relax some of these assumptions (in particular the assumption of independence of sources).
References: