Recognition of incomplete sequences using Fisher scores and hidden Markov models

V E Uvarov, A A Popov and T A Gultyaeva

Faculty of Applied Mathematics and Computer Science, Novosibirsk State Technical University, Novosibirsk, Russia

Abstract. We propose a method for recognition of incomplete sequences which makes use of feature vectors built from fisher scores for log likelihood function of hidden Markov model. We use support vector machines method to classify the feature vectors. The proposed method was compared to a previously developed method for incomplete sequence recognition based on marginalization of missing observations which can be considered as a state of the art method. The proposed method proved to be more effective than the SOTA method in situations when the percent of missing observations in training and testing sequences is high (more than 20% in our experiment). Thus, we suggest using the proposed method in situations when big percent of data is missing but the recognition still must be done.

1. Introduction

Hidden Markov model (HMM) conception was presented in 1970-s [1]. However, problems which concern using HMMs in case of incomplete data remain are poorly investigated. These problems are quite relevant since in complex systems, e.g. when receiving signals from spacecrafts or aircrafts, one has to deal with datastreams of various sources in noisy environments when there is a high possibility of data loss or corruption. In this paper, we deal with the problem of missing observations in sequences. From now on we will refer to such sequences as incomplete. We consider a case when such missing observations are not generated by random process itself but rather occur randomly in sequences because of some external interference.

2. Hidden Markov model

2.1. Structure of hidden Markov model

Hidden Markov model (HMM) describes a random process which appears to be in one of the $N$ hidden states at each time $t \in \{1, \ldots, T\}$ and transits to another or to the same state according to some transition probabilities. The states are hidden from the observer however they can be inferred from the observed sequences. In this paper, we consider HMM with continuous multivariate observation density, namely a mixture of multivariate normal distributions (often such HMM is called Gaussian HMM or GHMM).

We shall denote a hidden state of GHMM at time $t$ as $q_t$, observation at time $t$ as $o_t$ and observation without specific time as $o$. GHMM can be specified by initial state distribution $\Pi = \{\pi_i = p(q_1 = s_i), \ i = \overline{1, N}\}$, transition probabilities matrix $A = \{a_{ij} = p(q_{t+1} = s_j | q_t = s_i), \ i, j = \overline{1, N}\}$ and conditional multivariate distributions $B = \{b_i(o) = f(o | q = s_i), i = \overline{1, N}, o \in \mathbb{R}^d\}$. Here conditional multivariate distributions are mixtures of multivariate normal distributions $b_i(o) = \sum_{m=1}^{M} \tau_{im} g(o; \mu_{im}, \Sigma_{im}), i = \overline{1, N}, o \in \mathbb{R}^d$, where $M$ - number of mixture components for each hidden state, $\tau_{im} \geq 0$ - weight of $m$-th mixture component in $i$-th hidden state ($\sum_{m=1}^{M} \tau_{im} = 1$, $i = \overline{1, N}$), $\mu_{im}$ - mean of normal distribution from $m$-th component of $i$-th hidden
state, $\Sigma_{im}$ – covariance matrix of normal distribution from $m$-th component of $i$-th hidden state and
g($\alpha; \mu_{im}, \Sigma_{im})$, $\alpha \in \mathbb{R}^2$ – multivariate normal probability density function, i.e.
g($\alpha; \mu_{im}, \Sigma_{im}) = \frac{1}{\sqrt{(2\pi)^{2/2} |\Sigma_{im}|}} e^{-0.5(\alpha-\mu_{im})^T \Sigma_{im}^{-1}(\alpha-\mu_{im})}$, $\alpha \in \mathbb{R}^2$. Thus, some specific GHMM can be specified by a set of parameters $\lambda = \{\Pi, A, B\}$ [2].

2.2. Recognition of complete sequences using HMM

The final aim of recognition of a sequence $O$ using HMMs is to refer this sequence to one of the classes which correspond to random processes with indexes $1, D$ described by HMMs $\lambda_1, ..., \lambda_D$. Usually a maximum likelihood (ML) criterion is used for choosing the process which generated the sequence with the highest probability. In that case sequence $O$ is referred to a class $r^*$ for which the loglikelihood function is maximal, i.e. $r^* = \arg \max_{r \in \{1,...,D\}} \ln p(O | \lambda_r)$.

To calculate the loglikelihood function of sequence $O$ being generated by random process corresponding to model $\lambda$ one usually uses forward-backward algorithm [3]. To calculate $\ln p(O | \lambda)$ one needs only the forward part of the algorithm so we will describe only it. When working with probabilities (values which usually are very close to zero) it is highly recommended to use a scaling procedure to prevent underflow. Effective scaling method is described in [2].

The scaled forward part of the forward-backward algorithm computes scaled forward variables $p(\{\alpha_0, \alpha_1, ..., \alpha_T\}, q_t = s_i | \lambda)$, $t = 1, T$, $i = 1, N$, i.e. probabilities of sequence $\{\alpha_0, \alpha_1, ..., \alpha_T\}$ being generated by random process described by model $\lambda$ and that this process was in state $s_i$ at time $t$.

Algorithm that computes scaled forward variables and loglikelihood function:

1) initialization:
$$
\tilde{\alpha}_i(i) = \pi_i b_i(\alpha_i), i = 1, N;
$$

2) induction:
$$
\tilde{\alpha}_{t+1}(i) = b_i(\alpha_{t+1}) \left[ \sum_{j=1}^{N} \alpha'_j(j) a_{ji} \right], i = 1, N, t = 1, T-1,
$$

where
$$
\alpha'_j(j) = \frac{\tilde{\alpha}_i(j)}{\sum_{n=1}^{N} \tilde{\alpha}_i(n)}, j = 1, N, t = 1, T-1.
$$

We introduce scaling coefficients:
$$
c_t = \left( \sum_{i=1}^{N} \tilde{\alpha}_i(i) \right)^{-1}, t = 1, T,
$$

hence:
$$
\alpha'_i(i) = c_t \tilde{\alpha}_i(i), \quad i = 1, N, \quad t = 1, T-1,
$$
$$
\alpha'_i(i) = \left( \prod_{t=1}^{T} c_t \right) \alpha_i(i), \quad i = 1, N, \quad t = 1, T-1.
$$

Thus, loglikelihood function for a sequence $O$ and HMM $\lambda$ can be computed as follows:
$$
\ln \left[ p(O | \lambda) \right] = - \sum_{t=1}^{T} \ln c_t.
$$

3. Recognition of incomplete sequences using HMM

Prior to recognition of incomplete sequences using HMMs it is necessary to estimate the parameters of HMMs corresponding to random processes, i.e. to train HMMs. However, in this article we do not consider HMM training on incomplete sequences. This problem was covered in our previous works [4,
5, 6] where marginalization of missing observations approach was applied to modify the Baum-Welch training algorithm. We shall define incomplete sequence as a sequence $O$ which contains some undefined (missing) observations. We denote missing observation as $\emptyset$. Thus, $O = \{ o_i \in R^t, t = 1, T \}$, $R' = R^t \cup \{ \emptyset \}$.

3.1. Recognition of incomplete sequences using marginalization of missing observations

This approach is based on maximum likelihood criterion. It is obvious that the probabilities $b_i(o_i), i = 1, N, t = 1, T$ in formulas (1)-(5) which are used for calculation of loglikelihood function cannot be evaluated if $o_i = \emptyset$ since the actual value of $o_i$ is unknown. To use these formulas for incomplete sequences one must define $b_i(\emptyset), i = 1, N$ to some reasonable values.

The idea of the proposed approach is to consider that any value from $R^t \cup \{ \emptyset \}$ could have been in place of a missing observation before the data corruption occurred [7]. Keeping this in mind we shall integrate $b_i(o_i), i = 1, N$ over all possible values of $\emptyset$:

$$b_i(\emptyset) = \int_R b_i(x)dx = 1, \quad i = 1, N.$$ 

This result can be proven by the fact that only one observation $x$ can be generated by a random process at a time and that $b_i(x)$ is the conditional probability density of observation $x$ generation when the random process was at hidden state $s_i, i = 1, N$. Using the same idea, we shall define probability density of multivariate normal distribution from the mixtures for missing observation [7]:

$$g(\emptyset, \mu_m, \Sigma_m) = \int_R g(x, \mu_m, \Sigma_m)dx = 1, \quad i = 1, N, \quad m = 1, M.$$ 

Now probabilities $b_i(o_i), i = 1, N, t = 1, T$ are defined for all the values of $o_i \in R'$ hence formulas (1)-(5) can be extended for the case of incomplete sequences. The modified forward variables calculation algorithm (scaled) is presented below:

1) initialization:

$$\tilde{\alpha}_i(i) = \begin{cases} \pi_i, & o_i = \emptyset, \quad i = 1, N; \\ \pi b_i(o_i), & \text{otherwise}. \end{cases}$$

2) induction:

$$\tilde{\alpha}_{i+1}(i) = \begin{cases} \sum_{j=1}^N \alpha'_i(j)a_{ij}, & o_{i+1} = \emptyset, \quad i = 1, N; \\ b_i(o_{i+1}) \left[ \sum_{j=1}^N \alpha'_i(j)a_{ij} \right], & \text{otherwise}, \quad i = 1, T - 1. \end{cases}$$

where $\alpha'_i(j) = \frac{\tilde{\alpha}_i(j)}{\sum_{n=1}^N \tilde{\alpha}_i(n)}, \quad j = 1, N, \quad t = 1, T - 1$. The scaling coefficients and loglikelihood function are calculated by the original formulas (4) and (5).

We shall name the used approach as a “marginalization of missing observations” since here we marginalize out the random variable $x$ from distribution $b_i(x), i = 1, N$ for missing observations. It is easy to show that now we can perform incomplete sequence recognition using ML criterion because all the necessary formulas were extended to the case of missing observations. In our previous studies the marginalization approach outperformed all other strategies of dealing with missing values [5, 6].
3.2. Recognition of incomplete sequences using Fisher scores

The recognition of sequences using HMMs can be done not only with the maximum likelihood criterion. A method for recognition of complete sequences using Fisher scores and HMMs was previously investigated in our studies. This method showed an advantage over the ML criterion when HMMs of competing random processes were similar and when training and test sequences which were exposed to various noise conditions [8]. However, the case of completely missing observations was not included in that study. Since the missing observations can be interpreted as an effect of some noise conditions, it is advisable to investigate the applicability of this method to the recognition of incomplete sequences using HMMs which were trained on incomplete sequences.

For the sake of simplicity, we will consider the case of two-class classification with this method. Each sequence is described with a feature vector which consists of Fisher scores for a sequence and each of the competing HMMs. Let \( \{ O^1, O^2, \ldots, O^K \} \) be a set of training sequences and \( \lambda_1, \lambda_2 \) - two HMMs corresponding to two competing random processes. For each training sequence \( O \) from \( \{ O^1, O^2, \ldots, O^K \} \), a feature vector \( \left( \frac{\partial \ln p(O \mid \lambda_1)}{\partial \eta}, \frac{\partial \ln p(O \mid \lambda_2)}{\partial \eta} \right) \) is built. The transposed versions of these vectors are stacked together into a training matrix \( X \) where columns correspond to features (Fisher scores) and rows – to sequences. In addition, an answers vector is formed \( Y = \{ y_1, \ldots, y_K \} \), where \( y_k \in \{1,2\} \) is an index of HMM which corresponds to a random process that generated \( O^k \), \( k = 1, K \). In the next step a support vector machines (SVM) classifier is trained using training matrix \( X \) and answers vector \( Y \). To recognize a new sequence, a similar feature vector should be built and classified using the trained support vector machines classifier [9]. The described two-class problem can be easily generalized to a multiclass problem using “one-vs-all” or “one-vs-one” strategy widely used for binary classifiers.

Below we present formulas for calculation of Fisher scores for HMM [8]. Based on formula (5) we get:

\[
\frac{\partial \ln p(O \mid \lambda)}{\partial \eta} = \sum_{i=1}^{K} \left( \sum_{t=1}^{T} \frac{1}{\partial \eta} \frac{\partial c_{j}}{\partial \eta} \right). 
\]

(6)

To calculate the derivative of scaling coefficient with respect to some HMM parameter \( \eta \) one can use the following formula:

\[
\frac{\partial c_{j}}{\partial \eta} = c_{j} \sum_{i=1}^{N} \frac{\partial \alpha_{i}(i)}{\partial \eta}, t = 1, T. 
\]

(7)

To calculate \( \frac{\partial \alpha_{i}(i)}{\partial \eta} \), \( i = 1, N \) we shall differentiate step-by-step the algorithm for calculation of scaled forward variables:

1 step:

\[
\frac{\partial \alpha_{i}(i)}{\partial \eta} = \frac{\partial \alpha_{i}(i)}{\partial \eta}, i = 1, N; 
\]

(8)

2 step:

\[
\frac{\partial \alpha_{i}(i)}{\partial \eta} = \left[ \sum_{j=1}^{N} \left( \frac{\partial \alpha_{i-1}'(j)}{\partial \eta} a_{ij} + \alpha_{i-1}'(j) \frac{\partial a_{ij}}{\partial \eta} \right) \right] b_{i}(t) + \\
+ \sum_{j=1}^{N} \left( \alpha_{i-1}'(j) a_{ij} \frac{\partial b_{i}(t)}{\partial \eta} \right) ;
\]

(9)

where \( \frac{\partial a_{i-1}'(j)}{\partial \eta} = \frac{\partial c_{i-1}}{\partial \eta} \alpha_{i-1}'(j) + \frac{\partial \alpha_{i-1}(j)}{\partial \eta} c_{i-1}, i = 1, N, t = 2, T. \)
Thus, to calculate \( \frac{\partial \hat{\alpha}_i(t)}{\partial \eta}, \ i = 1, N \) we first need to calculate derivatives
\[
\frac{\partial \hat{\alpha}_i(t)}{\partial \eta}, \frac{\partial b_i(t)}{\partial \eta}, \frac{\partial a_{ij}}{\partial \eta}, \ i, j = 1, N, t = 1, T .
\]
To calculate derivative with respect to covariance matrix element in case of non-diagonal covariance matrix one must first differentiate the elements of inverse covariance matrix, hence here we will consider only diagonal covariance matrices \( \Sigma_{im} , \ i = 1, N, \ m = 1, M \).

Below we present formulas to calculate the derivatives \( \frac{\partial \hat{\alpha}_i(t)}{\partial \eta}, \frac{\partial b_i(t)}{\partial \eta}, \frac{\partial a_{ij}}{\partial \eta} \) for all the possible kinds of \( \eta \) parameter (which can be any of the following HMM parameters: \( \pi_i, a_{ij}, \tau_{im}, \mu_{im}, \Sigma_{im} \), \( i = 1, N, m = 1, M, z = 1, Z \)).

\[
\frac{\partial \hat{\alpha}_i(i)}{\partial \pi_j} = \begin{cases} b_i(1), \ i = j, i, j = 1, N, \\ 0, \ i \neq j \end{cases}, \quad (10)
\]

\[
\frac{\partial b_i(t)}{\partial \pi_j} = 0, \ i, j = 1, N, t = 1, T , \quad (11)
\]

\[
\frac{\partial \hat{\alpha}_i(i)}{\partial a_{ij}} = 0, \ i, i, j = 1, N, \quad (12)
\]

\[
\frac{\partial b_i(t)}{\partial a_{ij}} = 0, \ i, i, j = 1, N, t = 1, T , \quad (13)
\]

\[
\frac{\partial a_{ij}}{\partial x} = \begin{cases} 1, \ x = a_{ij}, \ i, j = 1, N, \\ 0, \ x \neq a_{ij} \end{cases}, \quad (14)
\]

\[
\frac{\partial b_i(t)}{\partial \tau_{im}} = \begin{cases} g(o_i; \mu_{im}, \Sigma_{im}), \ i = i, i, i = 1, N, m = 1, M, \\ 0, \ i \neq i \end{cases}, \quad (15)
\]

\[
\frac{\partial \hat{\alpha}_i(i)}{\partial \tau_{im}} = \begin{cases} \pi_i, \ \frac{\partial b_i(1)}{\partial \tau_{im}}, \ i = i, i, i = 1, N, m = 1, M, \\ 0, \ i \neq i \end{cases}, \quad (16)
\]

\[
\frac{\partial b_i(t)}{\partial \mu_{im}} = \begin{cases} 0.5 \tau_{im} g(o_i; \mu_{im}, \Sigma_{im}) \frac{\frac{a_i^2}{\Sigma_{im}} - \frac{\mu_{im}^2}{\Sigma_{im}}}{\Sigma_{im}} , \ i = i, \\ 0, \ i \neq i \end{cases}, \quad (17)
\]

\[
\frac{\partial \hat{\alpha}_i(i)}{\partial \mu_{im}} = \begin{cases} \pi_i, \ \frac{\partial b_i(1)}{\partial \mu_{im}} , \ i = i, i, i = 1, N, m = 1, M, \\ 0, \ i \neq i \end{cases}, \quad (18)
\]

\[
\frac{\partial b_i(t)}{\partial \Sigma_{im}} = \begin{cases} 0.5 \tau_{im} g(o_i; \mu_{im}, \Sigma_{im}) \left( \frac{\frac{a_i}{\Sigma_{im}} - \frac{\mu_{im}}{\Sigma_{im}}}{\Sigma_{im}} \right) - \frac{1}{\Sigma_{im}} , \ i = i, \\ 0, \ i \neq i \end{cases}, \quad (19)
\]
\[ \frac{\partial \alpha_t(i)}{\partial \Sigma_{im}} = \begin{cases} \frac{\partial \alpha_t(1)}{\partial \Sigma_{im}}, & i = i_i, \quad i, i_i = 1, N, m = 1, M, \quad z = 1, Z. \\ 0, & i \neq i_i \end{cases} \]  

(20)

Formulas (6)-(20) can be extended to the case of incomplete sequences using the marginalization of missing observations approach which was described in the previous subsection. Thus in (6)-(20) we consider \( b_t(\emptyset) = 1, \quad i = 1, N \) and \( g(\emptyset, \mu_m, \Sigma_m) = 1, \quad i = 1, N, m = 1, M \). In addition, we should make additional changes to formula (17):

\[ \frac{\partial b_t(t)}{\partial \mu_m'} = \begin{cases} 0.5 \tau_m g(o_t; \mu_m', \Sigma_m) \frac{o_t - \mu_m'}{\Sigma_m'}, & i = i_t \text{ and } o_t \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \]

\[ i, i_t = 1, N, \quad t = 1, T, \quad m = 1, M, \quad z = 1, Z, \]

and formula (19):

\[ \frac{\partial b_t(t)}{\partial \Sigma_{im}'} = \begin{cases} 0.5 \tau_m g(o_t; \mu_m', \Sigma_m) \left( \frac{o_t - \mu_m'}{\Sigma_m'} \right)^2 - \frac{1}{\Sigma_m'}, & i = i_t \text{ and } o_t \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \]

\[ i, i_t = 1, N, \quad t = 1, T, \quad m = 1, M, \quad z = 1, Z. \]

4. Evaluation

In this section, we compare methods for recognition of incomplete sequences: the proposed one that uses Fisher scores and HMM and state-of-the-art method based on maximum likelihood criterion.

For the experiment we used two HMMs \( \lambda_1 \) and \( \lambda_2 \) with the following parameters: the number of hidden states \( N = 3 \), number of mixture components \( M = 3 \), dimensionality of observations \( Z = 2 \).

Initial distribution vector \( \Pi = [1, 0, 0] \), transition probabilities matrix

\[ A = \begin{pmatrix} 0.1 + \Delta A & 0.7 - \Delta A & 0.2 \\ 0.2 & 0.2 + \Delta A & 0.6 - \Delta A \\ 0.8 - \Delta A & 0.1 & 0.1 + \Delta A \end{pmatrix}, \]

mixture components weights

\[ \{ \tau_m, i = 1, N, m = 1, M \} = \begin{pmatrix} 0.3 + \Delta \tau & 0.4 - \Delta \tau & 0.3 \\ 0.3 & 0.4 + \Delta \tau & 0.3 - \Delta \tau \\ 0.3 - \Delta \tau & 0.4 & 0.3 + \Delta \tau \end{pmatrix} \]

(here rows correspond to hidden states and columns – to mixture components), mean vectors of mixture components

\[ \{ \mu_m, i = 1, N, m = 1, M \} = \begin{pmatrix} (0 - \Delta \mu, 0 + \Delta \mu) \quad (1 - \Delta \mu, 1 + \Delta \mu) \quad (2 - \Delta \mu, 2 + \Delta \mu) \\ (3 - \Delta \mu, 3 + \Delta \mu) \quad (4 - \Delta \mu, 4 + \Delta \mu) \quad (5 - \Delta \mu, 5 + \Delta \mu) \\ (6 - \Delta \mu, 6 + \Delta \mu) \quad (7 - \Delta \mu, 7 + \Delta \mu) \quad (8 - \Delta \mu, 8 + \Delta \mu) \end{pmatrix} \]

(here rows correspond to hidden states and columns – to mixture components), all the covariance matrices of mixture components \( \{ \Sigma_m, i = 1, N, m = 1, M \} \) were diagonal with the same value on diagonal: \( 0.1 + \Delta \sigma \). The first model had \( \Delta A = 0, \quad \Delta \tau = 0, \quad \Delta \mu = 0, \quad \Delta \sigma = 0 \) and the second model had \( \Delta A = 0.05, \quad \Delta \tau = 0.05, \quad \Delta \mu = 0.01, \quad \Delta \sigma = 0.01 \). Such choice of parameters makes the recognition task extremely difficult since the competing random processes corresponding to the two HMMs have very similar properties and observations generated by them are difficult to distinguish. Using each of the \( \lambda_1 \) and \( \lambda_2 \) HMMs we generated \( K = 100 \) training and test sequences of length \( T = 100 \) each of which contained \( G \) missing observations (\( G \) was changed in range from 0 to 90 during the experiment) in
random places. Using incomplete training sequences, we estimated parameters of the models $\lambda_1$ and $\lambda_2$ using the modified Baum-Welch algorithm and marginalization approach [4, 5, 6]. In addition, we estimated parameters of support vector machines classifier with the approach described in subsection II.B and selected the best SVM hyperparameters using cross-validation on 4 folds on the training set. Then we used $\lambda_1$ and $\lambda_2$ to recognize the test incomplete sequences using the proposed method that uses Fisher scores and HMM (dashed line) and method based on maximum likelihood criterion (solid line). The recognition accuracy on test set is displayed on the y-axis and the percent of missing observations in training and test sequences is displayed on the x-axis in figure 1. The presented results are the average of 50 launches made with different initial seeds for random numbers generator.

![Figure 1. Recognition accuracy on the test set of the Fisher scores and ML criterion methods.](image)

The proposed Fisher scores method starts to outperform the ML criterion method roughly after 20% of missing observations in training and test sequences. The advantage of the proposed method raises proportionally to the percent of missing observations: it is up to 10% more accurate when 90% of data is missing.

5. Conclusion

We introduced a novel method to recognize incomplete sequences which is based on Fisher kernels and hidden Markov models assisted by support vector machines classifier. The proposed method was compared to the state-of-the-art method of incomplete sequence recognition which utilizes maximum likelihood criterion. The comparison showed that the proposed method provides more recognition accuracy than the SOTA method when the percent of missing data in sequences is large (more than 20% in our experiment) and when competing hidden Markov models’ parameters are close. Thus, the proposed method is recommended to be used in situations when there are a lot of missing data but the recognition of incomplete sequences still must be done to some extent. However, in normal situation when HMMs are not very close and the percent of missing data is not high we advise against using the proposed method since it gives additional computational overhead comparing to the SOTA method.

References

[1] Baum L and Petrie T 1966 Statistical inference for probabilistic functions of finite state Markov chains The Annals of Mathematical Statistics 37 pp 1554-1563
[2] Rabiner L 1989 A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition Proceedings of the IEEE 77 pp 257-285
[3] Baum L and Egon J 1967 An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology Bulletin of the American Meteorological Society 73 pp 360-363
[4] Popov A, Gultyaeva T and Uvarov V 2016 Training hidden Markov models on incomplete sequences *Proceedings of Russian scientific conference Information processing and mathematical modelling* pp 125-139

[5] Popov A, Gultyaeva T and Uvarov V 2016 A Comparison of Some Methods for Training Hidden Markov Models on Sequences with Missing Observations *Proceedings of 11th International Forum on Strategic Technology IFOST-2016* 1 pp 431-435

[6] Popov A, Gultyaeva T and Uvarov V 2016 Training hidden Markov models on sequences with missing observations *Proceedings of 13th International Conference on Actual Problems of Electronic Instrument Engineering (APEIE 2016)* 1 pp 317-320

[7] Cooke M, Green P, Josifovski L and Vizinh A 2001 Robust automatic speech recognition with missing and unreliable acoustic data *Speech Communication* 34 pp 267-285

[8] Gultyaeva T, Popov A, Kokoreva V and Uvarov V 2015 Classification of observation sequences described by Hidden Markov Models *Proceedings of the International Workshop Applied Methods of Statistical Analysis Nonparametric approach AMSA-2015* pp 136–143

[9] Boser B, Guyon I and Vapnik V 1992 A training algorithm for optimal margin classifiers *Proceedings of the fifth annual workshop on Computational learning theory* p 144