The Classification of Noisy Sequences Generated by Similar HMMs

A.A. Popov and T.A. Gultyaeva

Department of Software and Database Engineering,
Novosibirsk State Technical University, Russia
alex@fpm.ami.nstu.ru, gult_work@mail.ru

Abstract. The method for classification performance improvement using hidden Markov models (HMM) is proposed. The k-nearest neighbors (kNN) classifier is used in the feature space produced by these HMM. Only the similar models with the noisy original sequences assumption are discussed. The research results on simulated data for two-class classification problem are presented.

Keywords: Hidden Markov Model, Derivation, k Nearest Neighbors.

1 Introduction

HMM are a powerful tool for modeling of various processes and pattern recognition. By their nature, Markov models allow to deal with spatial-temporal sequence characteristics directly and therefore they became widely-used [1], [2], [3]. However, despite a wide circulation of models of this kind, HMM possess low enough classification abilities. Though these models are widespread HMM possess of sufficiently low classification properties. At the same time it is well known that the HMM have a fairly low classification ability.We understand the classification problem by the following. We have an object set discriminated on classes by expert (training with teacher). This object set is named the training set. We want to create an algorithm that will be able to classify a test object from origin set. The two-class classification problem using the matrix of distances between the objects is discussed in this article. In this case, the each object is being described by distances to all other objects from training set. The method of the nearest neighbors, the Parzen windows method and the method of potential functions work with input data of such type. The set of Gaussian time sequences generated by two HMMs with similar parameters are taken as classification objects. In order to approximate real-world examples all observed time sequences being distorted. The task was to compare the capabilities of traditional methods of classification with a simple nearest neighbor classifier [4] in the space of first derivatives of the likelihood function for the HMM parameters.

The remainder of this article is organized as follows. Section 2 introduces the method of solution of assigned task. Section 3 provides some results of computational experiments. Section 4 summarizes our findings.

S.O. Kuznetsov et al. (Eds.): PReMI 2011, LNCS 6744, pp. 30–35, 2011.

2 The Method of Sequences Classification

An HMM is completely described by the following parameters:

1. The initial state distribution $\mathit{\Pi} = \{\pi_i\}, i = \overline{1, N}$, where $\pi_i = P\{q_1 = i\}$ and N – is the number of hidden states in the model.
2. The matrix of state transition probabilities $A = \{a_{ij}\}$, where $a_{ij} = P\{q_t = j | q_{t-1} = i\}$, $i, j = \overline{1, N}$, $t = \overline{1, T}$, where T – is the length of observable sequence.
3. The matrix of observation symbols probabilities $B = \{b_{ij}\}$, $i = \overline{1, N}$, $j = \overline{1, M}$, where $b_i(j) = P\{o_t = v_j | q_t = i\}$, o_t – is the symbol observed at the moment of time $t = \overline{1, T}$, M – is the number of observation states in the model. In this work the case when function observable symbol probabilitie distribution is described by a mix of normal distributions is considered in such a manner that the one hidden state is associated with one observable state: $b_i(t) = (\sqrt{2\pi}\sigma_i)e^{-(o_t - \mu_i)^2/2\sigma_i^2}$,$i = \overline{1, N}$, $t = \overline{1, T}$.

Thus, HMM is completely described by the matrix of state transition probabilities, the probabilities of observation symbols and the initial state distribution: $\lambda = (A, B, \pi)$.

A classifier based on log-likelihood function is traditionally used. The sequence O is considered as being generated by model λ_1 if (1) is satisfied:

$$lnL(O|\lambda_1) > lnL(O|\lambda_2). \tag{1}$$

Otherwise, it is considered that the sequence is generated by model λ_2.

Some authors (e.g. [5],[6]) propose to use the spaces of the so-called secondary features as a feature space in which the sequences are being classified. For example, forward-probabilities and backward-probabilities, which are used for computation of probability that the sequence is generated by model λ, can be used as a secondary features. The first derivatives of space of likelihood function logarithm are also used. These derivatives are being taken with respect to different model parameters. The authors [5] offer to include the original sequence into the feature vector also. In this work the performance of two-class classification in the space of the first derivatives of the likelihood function is discussed.

The classification problem states as follows. There are two groups of training sequences: the first group consists of sequences generated by λ_1, and the second group – by λ_2. Usually, in order to determine which class the test sequence O^{test} is belongs to the rule (1) is used. Because the model parameters λ_1 and λ_2 are unknown, at first one needs to estimate them (for example, the algorithm of Baum-Welch is used for it), and then calculate them according to rule (1).

If the competing models have similar parameters, and the observed sequences are not purely Gaussian sequence, the traditional classification technique using (1) does not always give acceptable results.

The following schema that increases discriminating features of HMM.

Step 1. For each training sequence $O_l^{learn_i}$, $i = \overline{1, K_l}$, $l = \overline{1, 2}$ where K_l – the count of training sequences for class with number l, the characteristic vector which can consist of all or a part of the features is being formed. The likelihood function is being calculated as for true class model to which training vectors are belong to as for model of other class. As a result the characteristic vector for the training sequence $O_l^{learn_i}$ generated by model λ_1 consists of two subvectors: $V_l^{learn_i} = (\, Z(O_l^{learn_i}, \lambda_1), \qquad Z(O_l^{learn_i}, \lambda_2)\,)^T$, where the first subvector consists of features, initiated by the model λ_1 , and the second – by the model λ_2.

Step 2. Similarly, the characteristic vector is calculated for the test sequence O^{test}.

Step 3. Using a metric based classifier (e.g. kNN) it is become clear to which class O^{test} belongs to.

3 Computing Experiments

Investigations were performed under following assumptions. The models λ_1 and λ_2 are defined on the hidden Markov chains with identical and they have differences in the matrix of transition probabilities only. For the first model λ_1 and for the second model λ_2 the difference was in transitive probabilities only:

$$\mathbf{A}_{\lambda_1} = \begin{pmatrix} 0.1 \; 0.7 \; 0.2 \\ 0.2 \; 0.2 \; 0.6 \\ 0.8 \; 0.1 \; 0.1 \end{pmatrix}, \mathbf{A}_{\lambda_2} = \begin{pmatrix} 0.2 \; 0.6 \; 0.2 \\ 0.2 \; 0.3 \; 0.5 \\ 0.7 \; 0.1 \; 0.2 \end{pmatrix}.$$

The Gaussian distribution parameters for the models λ_1 and λ_2 are chosen identical: $\mu_1 = 0$, $\mu_2 = 5$, $\mu_3 = 10$, $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 1$. The probabilities of initial states are coincide also: $\pi = (1, 0, 0)$. Thus, models have turned out very close to each other, and, hence, sequences differ among themselves very weakly.

The training and testing sequences have been simulated by the Monte-Carlo method. It has been generated 5 training sets of 100 sequences for each of the classes to perform investigations. For each training set 500 test signals were generated for each class. The results of classification have been averaged. The length of the each sequence has been set to 100.

3.1 The Additive Noise

The first variant of distortion of a true sequence assumed additive superposition of the noise component distributed under some distribution law. We denoted the sequence simulated on model λ through u. Then at superposition of noise e on this sequence according to the following formula we received noisy sequence with an additive noise: $y = (1 - \omega)u + \omega e$, where ω shows the influence of sequence distortion.

The space of the first derivatives of likelihood function with respect to elements of transition probabilities matrix has been chosen as the feature space for the kNN classifier. Further in Tables 1–3 following designations are used: APD – the average percent of difference between the results of the kNN classifier and

Table 1. The comparison of classification's results at the additive noise

	$e \succ N(0, 25)$		$e \succ C(0, 0.01)$	
ω	APD	AP	APD	AP
0.1	-1.5	89.12	11.18	88.56
0.2	-1.48	88.98	7.86	84.86
0.3	-2.14	85.82	22.62	84.26
0.4	-1.64	80.58	2.4	74.14
0.5	-0.52	72.28	-1.88	72.44
0.6	-4.28	58.98	-3.44	68.08
0.7	-2.66	53.18	-3.08	67.18
0.8	-2.82	49.26	-0.12	49.76
0.9	-1.44	49.88	0.92	50.52

the results of traditional classification; AP – the average percent of correctly classified sequences using the kNN classifier.

The classification results with the normal noise distribution are shown in Table 1 in the 2nd and the 3rd columns. As follows from this experiment, the kNN classifier using the space of the first derivatives of likelihood function gives worse results than traditional classifier. It is explained by the fact that the noises and sequences have identical normal distribution, and the algorithm of Baum-Welch being used for parameters estimation is exactly tuned for parameters estimation of probability distribution function of observed sequences in the conditions when this function is the normal distribution function. The results of classification comparison at the noise distributed on the Cauchy distribution law are in Table 1 in the 4th and the 5th columns. In this case there is opposite situation: the kNN classifier gives better results at the noise level $\omega \leq 0.4$.

3.2 Probability Substitution of a Sequence by Noise

The second variant of distortion of a true sequence assumed partial substitution of a sequence by noise under the probability scheme, i.e. with some probability p instead of the true sequence associated with some hidden state, the noise sequence was appearing. At this time the parameters of noisy sequence were varying in the different experiments.

The results of classification comparison at the noise distributed on the normal distribution law are in Table 2 from the 2nd to the 5rd columns. In the 2nd and the 3rd columns as a result of superposition of such noise there was a displacement of the estimated parameters of expectation, but the traditional classifier showed better results than the proposed one. In the next two columns there is stable classification results improvement when the probability of noise appearance $p \leq 0.6$. It is explained by the fact that the parameters of distribution of noise generator are very big values unlike the previous case. The results of classification comparison when the noise has a Cauchy distribution are shown in

Table 2. The comparison of classification's results at the probability substitution of a sequence by noise that hasn't dependence on the hidden state

	$e\succ\!\!-N(-5,1)$		$e\succ\!\!-N(30,100)$		$e\succ\!\!-C(0,0.1)$	
p	APD	AP	APD	AP	APD	AP
0.1	-2.44	81.88	21.36	78.28	12.32	82.1
0.2	-2.96	78.38	3.96	52.88	17.4	76.76
0.3	-4.6	71.04	3.74	51.99	21.4	73.68
0.4	-2.62	69.84	6.46	52.61	6.42	61.32
0.5	-2.48	65.32	1.32	51.62	3.42	54.2
0.6	-2.45	59.81	2.84	52.72	2.58	52.22
0.7	-0.8	55.54	-1.3	50.85	0.71	50.49
0.8	-2.54	52.46	1.76	50.14	1.66	51.64
0.9	-0.1	50.1	0.04	50.24	1.09	50.83

Table 2 in the 6th and the 7th columns. In this case there is stable advantage of the kNN classifier based classifier.

Table 3. The comparison of classification's results at the probability substitution of a sequence by noise that has dependence from the hidden state

	$e\succ\!\!-N$		$e\succ\!\!-C$	
ω	APD	AP	APD	AP
0.1	-0.58	79.1	11.46	84.06
0.2	2.78	70.54	17.08	80.42
0.3	4.3	65.52	13.66	63.04
0.4	3.78	57.2	3.17	53.95
0.5	3.8	56.94	1.74	52.2
0.6	2.96	57.08	0.81	51.11
0.7	0.12	60.94	0.14	50.18
0.8	-0.6	70.04	0.85	51.53
0.9	-2.27	77.35	1.36	50.98

Classification results with the normal noise distribution: $e_1\succ\!\!-N(10,1)$, $e_2\succ\!\!-N(0,1)$, $e_3\succ\!\!-N(5,1)$ (where e_i, $i=\overline{1,3}$ – it is the noise appearing when the HMM is in the ith hidden state) are shown in Table 3 in the 2nd and the 3rd columns. The observed sequences have the double-mode distribution instead of single-mode distribution expected at $p \in [0.2;0.7]$. The traditional classifier is slightly worse than the proposed kNN classifier in this situation. The results of classification comparison at the noise distributed on the Cauchy distribution law: $e_1\succ\!\!-C(0,1)$, $e_2\succ\!\!-C(5,1)$, $e_3\succ\!\!-C(10,1)$, are in Table 3 in the 4th and the 5th columns. In this case the noise substituting the original sequences is differed from the last by using distribution law of random variables only. In this experiment it is observed the constant advantage of the offered method that used the kNN classifier. Similar results were obtained in the noise parameters distributed by Cauchy distribution law: $e_1 \succ C(0,1)$, $e_2 \succ C(0,0.5)$, $e_3 \succ C(0,0.1)$, i.e. with the absence of noise displacement but with different scale of distribution.

4 Conclusion

In this article it was shown that the feature space generated by the HMM can be used for the classification of sequences generated by the similar models. The first derivatives of likelihood function logarithm with respect to the parameters of HMM were used as the features. The kNN classifier was used in this feature space. Studies have shown that with similar models and signals with the distortion the proposed method can improve the quality of classification.

References

1. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE 77(2), 257–285 (1989)
2. Cappé, O.: Ten years of HMMs. CNRS, LTCI & ENST, Dpt. TSI, `http://perso.telecom-paristech.fr/~cappe/docs/hmmbib.html`
3. Mottl, V.V., Muchnik, I.B.: Hidden Markov Models in Structural Signal Analysis Moscow, Russia (1999) (in Russian)
4. Zagorujko, N.G.: Applied methods of analysis of data and knowledge. Novosibirsk, Russia (1999) (in Russian)
5. Chen, L., Man, H.: Combination of Fisher Scores and Appearance Based by Features For Face. In: Proc. of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, Berkeley, California, USA, pp. 74–81 (2003)
6. Aran, O., Akarun, L.: Recognizing Two Handed Gestures with Generative, Discriminative and Ensemble Methods Via Fisher Kernels. In: Gunsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 159–166. Springer, Heidelberg (2006)