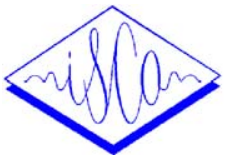


EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

K-NN VERSUS GAUSSIAN IN HMM-BASED RECOGNITION SYSTEM

Claude Montacié, Marie-José Caraty and Fabrice Lefèvre

LIP6 - Université Pierre et Marie Curie - CNRS
4, place Jussieu - 75252 Paris Cedex 5 - France
Tel. (33/0) 1 44 27 62 81, FAX (33/0) 1 44 27 70 00, e-mail: montacie@laforia.ibp.fr

ABSTRACT

For many years, the K-Nearest Neighbours method (K-NN) is known as one of the best probability density function (pdf) estimator. A fast K-NN algorithm has been developed and tested on the TIMIT database with a gain in computational time of 99;8%. The K-NN decision principle has been assessed on a frame by frame phonetic identification. A method to integrate K-NN estimator pdf in a HMM-based system is proposed and tested on an acoustic-phonetic decoding task. Finally, preliminary experiments are performed on the HMM topology inference .

1. INTRODUCTION

In continuous HMM, it is usual to represent the state output distribution by a gaussian mixture density. The distribution of the observations (*i.e.*, the speech analysis vectors) is represented by a weighted sum of gaussian probability densities. But analysis vectors have not generally a gaussian distribution [2]. The choice of the number of mixtures is generally guided by heuristics. For any parametric probability density function (e.g., laplacian), representing the state output distribution, the same problem occurs. We have chosen to develop a HMM-based system using the non-parametric K-NN estimator. This estimator is well known for its asymptotic performance (*i.e.*, with a large amount of data). The 1-NN error is lower than twice the error of the best estimator (*i.e.*, Bayesian estimator). The K-NN error decreases when K increases. The difficulty raised by the huge computational cost of this estimator can be solved by the development of Fast K-NN algorithms.

2. FAST K-NN ALGORITHMS

Three methods have been used to obtain a very fast K-NN algorithm : the Friedman and the Fukunaga methods. Their principle are introduced.

-The Kittler-Richetin method [3,4] estimates the euclidian measure (L_2) using another lower-cost measure such as the Manhattan measure (L_1) or the Max measure (L_∞). It is based on the inequalities (1) and (2) between L_1, L_∞ and L_2 measures:

$$\sqrt{d}\, L_2(v_1, v_2) \geq L_1(v_1, v_2) \geq L_2(v_1, v_2)$$

$$L_2(v_1, v_2) \geq L_\infty(v_1, v_2) \geq \frac{L_2(v_1, v_2)}{\sqrt{d}}$$

where v_1 and v_2 are d-dimensional vectors.
The gain in computational time for this method is about 80%.

-The Friedman method [5] searchs the K-NN using at first the components of maximum dispersion. Its gain in computational time is about 75%.

-The Fukunaga method [6] splits up the training analysis vectors into hierarchical clusters. The K-NN computation of a vector v_1 is processed by a depth-first exploration of the hierarchical structure. Each cluster Sp is studied, its centroid Cp and radius Rp are computed. Two inequalities (3) and (4) are used to avoid useless computations. If the inequality (3) is true, no vector of Sp will be one of the K-NN of the vector v_1. If the inequality (4) is true the vector y is not one of the K-NN of the vector v_1.

$$L_2(v_1, C_p) \geq d_K + R_p$$

$$L_2(v_1, C_p) \geq d_K + L_2(v_2, C_p)$$

where d_K is the measure between the vector v_1 and the K^{th} nearest vector among the last treated vectors.
The gain in computational time is about 96%.

The fast K-NN algorithm, we have developed from these methods, has been tested on the TIMIT speech database. The training set is composed of 1,124,823 frames (3,696 sentences), the core-test is composed of 57,919 frames (192 sentences). Computed per centi-second, a frame is represented by 12 MFCC and by the energy coefficient. A decomposition in 30,000 clusters has been used. For each vector of the training and test-set, the K-NN are computed from its 50 nearest vectors in the training set. The gain in computational time is about 99.8% (four 50-NN estimations per second).

3. K-NN ASSESSMENT

Two preliminary experiments aim at the assessment of the K-NN estimator. The experiments are carried out on the 57,919 frames of the core-test TIMIT. The reference labeling of TIMIT is used [7].

In the first experiment, for each test-vector, we compute the number k of its 50-NN (in the training set) of same phonetic label. The figure 1 gives for a given k, the percentage of test-vectors having k-NN of same label.

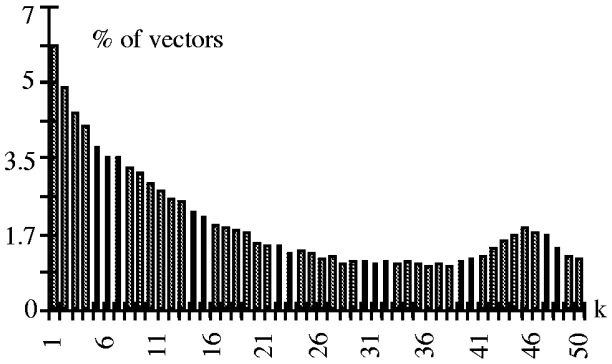


Figure 1. Histogram acording to the value k of the vector percentage having k-NN of same label

In average, a test-vector has 18 nearest neighbours of the same phonetic label. Three distinctive parts are distinguished. The first one (k<18) represents the vectors far from their proper training set. These vectors have to be studied to analyse precisely the reasons of such spacial distortions. The second part (k>25) represents the vectors a priori easy to identify, as for instance using the majority votation. The last part (18≤k≤25) represents the vector which identification is possible using a more sophisticated principle.

In the second experiment, for each phonetic label, the identification frame rate has been computed using the 50-NN maximum maximorum decision principle.

Phon.	aa	ae	ah	aw	ay	b	ch	d	dh
Ident. %	55	34	27	01	13	02	01	01	04
Phon.	dx	eh	er	ey	f	g	h#	hv	ix
Ident. %	02	17	57	26	64	01	91	12	55
Phon.	iy	jh	k	l	m	n	ng	ow	oy
Ident. %	62	04	30	52	35	53	07	17	00
Phon.	p	q	r	s	sh	t	uh	ux	v
Ident. %	17	02	21	84	67	23	00	15	23
Phon.	w	y	z						
Ident. %	36	01	23						

Table 1. Phonetic identification frame rate

For this frame by frame phonetic decoding, the average of the identication rate is about 50 %. Three reasons can explain this result. The first one is the low occurrence number of phonemes such as [ch, dh, jh, ng, uh, y]. The second one is the segmentation error inherent to any labeling. The third one is the difficulty to identify complex phonemes such as diphtongs or plosives using a single frame. Therefore, it is necessary to integrate the K-NN information in a global decision principle.

4. K-NN ESTIMATOR IN HMM

Training and decoding techniques for HMM do not rely on the used pdf. In practice, some adaptations are required from a gaussian pdf-based system to a K-NN pdf-based system. We have adapted the Forward Backward (FB) and Viterbi algorithms in HMM ToolKit 1.4 [8] to handle the K-NN estimator.

4.1 Adaptation Principle

The state output probabilty calculation in HMM is adapted to K-NN estimator. For this, we compute for each training vector its state occupation probabilities.
The state occupation probability of state s for the vector v of the vector sequence V is computed as :

$$P^{s}_{Occ}(v) = \frac{P(V, X(v) = s|M)}{P(V|M)}$$

where X is a state sequence of V, and M the model of V accordingly to a previous or reference alignment.
Both numerator and denominator probabilities are derived from FB or Viterbi algorithm. The Viterbi algorithm finds the maximum likelihood state sequence, so each observation vector is assigned to a single state. That is, P^{s}_{Occ} is one for one state and zero for the others. Whereas, in FB, the full likelihood is computed on all possible state sequence and each observation vector is assigned to every state in proportion of the model being in that state when the vector was observed.
Thereafter, the state output probability of any vector is calculated as the normalized summation of the state occupation probabilities of its K-NN :

$$P^{s}_{Out}(v) = \frac{\sum_{k=1}^{K} P^{s}_{Occ}(k^{th} - NN(v))}{K}$$

where k^{th}-$NN(o)$ is the k^{th} nearest neighbour of observation vector v.
We have introduced this state output probability calculation in the FB and Viterbi algorithms. Usually, the Viterbi algorithm is used to perform a first estimation before the re-estimation with FB. Unfortunately, in our case, this first estimation with Viterbi does not fit since nearly all the vectors are finally assigned to one single state. The direct assignement of vectors to individual states in Viterbi creates an irreversible accumulation phenomenon. To avoid this, an original method is proposed to estimate the models without a first Viterbi stage.

4.2 HMM Estimation Improvement

A HMM training assumes initial estimates of the HMM parameters. Commonly, a rough guess of the initial pdf

values is obtained by an uniform segmentation of the observations sequences, associating each successive segment with successives states. Then a first estimation is made with Viterbi algorithm, which we saw is not adapted in our case. To address this, the first estimation of the models is obtained from a new method using the information provided by the K-NN.
The parameters are initialized by a uniform segmentation. Then, a time to states projection is performed. That is, each vector is associated to every states proportionnally to the number of its K-NN in this state considering the uniform segmentation. Then, the vector sequences are divided into time-proportionnal clusters. So as the time/states projections are averaged amongst all vectors of each cluster. These average time/states projections are used to initialize state occupation probabilities.

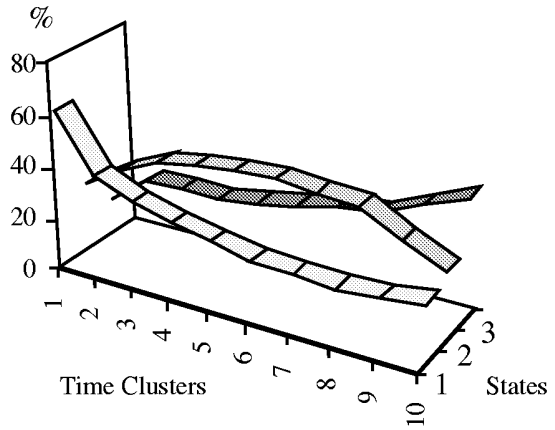


Figure 2. Time-states projection for vowel /uw/.

Figure 2 illustrates the time/states projection in a three-states Bakis model for the vowel /uw/, with 50-NN. For instance, the state occupation probabilities of the observations of the fifth cluster (*i.e.*, comprised within 40 and 50% of their sequence duration) are 0.26 for the first state, 0.46 for the second one and 0.27 for the third one. The time/states projections of all phonetic classes confirm the relation between spacial and temporal proximities and thus the use of Bakis-kind model.
The experimental trainings performed with the time/states projection do not show any significant gain in the accuracy of the model estimation (*i.e.*, do not increase the average model likelihood). Beside, the FB re-estimation convergence is reached faster than with a simple uniform segmentation initialisation.

5. EVALUATIONS

The experiments aim at comparing the gaussian and the K-NN estimators in HMM-based system. Two kinds of assessments are used : the recognition rate of the acoutic-phonetic decoding and the Segmental Normalised Acoustic Likelihood Coefficient (SNALC).

5.1 Acoustic-phonetic decoding.

The choosen task is the reference [9] acoustic-phonetic decoding on TIMIT database core-test. The basis model is a three-states Bakis. The gaussian distribution is a 8-gaussian multivariate mixture with the frame representation previously described. Both systems use a phonetic back-off bigram learned on the trainset.
Table 2 presents the recognition results for the 8-gaussian and 50-NN systems.

	%Corr	%Acc.	Del.	Subs.	Ins.
50-NN	58.17	51.84	14.91	26.91	6.33
8-gaussian	52.35	50.06	52.35	23.56	2.29

Table 2. Recognition results for TIMIT core-test.

These results are low but related to the simplicity of the models involved. The gain in correct recognition rate with the 50-NN estimator is nearly 6%. This difference is lowered to 1.8% in accuracy rate due to the high level of insertion in the 50-NN system (6.3%). The confidence interval of 95% for an estimate mean of 50% upon 7,215 units is 1.1%. Thus the difference between accuracy rates is barely statistically meaningful. We propose to compare thoroughly the systems with the SNALC.

5.2. SNLAC Evaluation

The SNALC is an attempt to a better evaluation of the pdf influence in a decision process such as HMM. This coefficient is computed for each frame during a Viterbi decoding as :

$$SNALC = 1 - \frac{L_{ref}/T_{ref}}{\sum_i L_i/T_i}$$

where L_i is the i^{th} model likelihood for a segment of T_i frames. The reference phoneme corresponds to the TIMIT alignement.

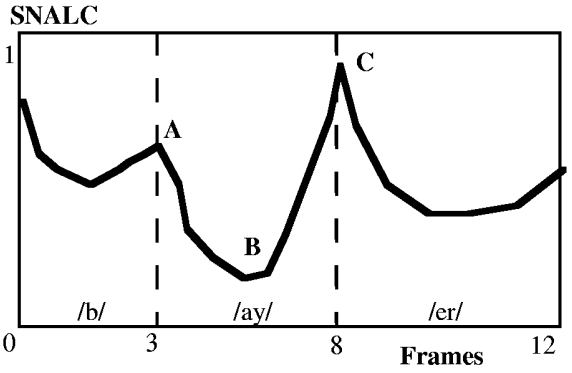


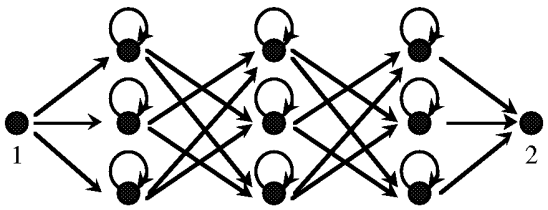
Figure 3. Typical behaviour of SNALC.

The figure 3 is an illustration of a typical behaviour of SNALC, here on an utterance of the world "buyer". From the beginning of the phoneme /ay/ (A), SNALCs decrease to B : the /ay/ model likelihood becomes preponderant. Then, the likelihood of /er/ grows up and the SNALCs raise up to the end of /ay/ (C). The SNALC values are relevant to confusions between phonemes or co-articulation phenomena. Thus, the average SNALC can be used as an assessment measure.

Experiments were performed on the TIMIT core test in the conditions described in 3. The average SNALC for K-NN (0.62) is lower than the gaussian one (0.74). SNALC shows the K-NN estimator interest.

6 TOPOLOGY INFERENCE

The point adressed here is the extension of HMM training techniques to the learning of the HMM topology (number of states, transitions between states) from the observations. Some attempts have already been made in topology inference [10, 11, 12]. The K-NN estimator pdf in HMM could be an asset in the issue of a posteriori topology inference.
A paradigm for the a posteriori topology inference is the training of an ergodic, fully connected, model. The probabilities of the useless state-transitions would be reduced to near zero. After removing of its unused parts, such model should have an optimal structure relevant to its complexity. In practice, the avalaible data limit the structural complexity of usable models.
Considering this limitation, we propose an initial structure (Figure 4), compromise between complexity and trainability.



(1 and 2 are non-emiting states)

Figure 4. A parallalized three-states Bakis model.

This model can be viewed as a parallelized version of three-states Bakis models. The use of three-states models is relevant to the supposed different intervals of a phone, left (on-glide phase), middle (central-glide) and right (off-glide phase).
Initialisation of the model is done from three spacial clusters of observation vectors. A state row is attributed to each cluster. Each vector is assigned to its cluster row states with the equiprobability of the state occupations.

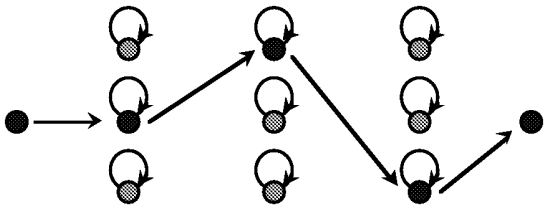


Figure 5. Model of vowel /uw/ after reestimation.

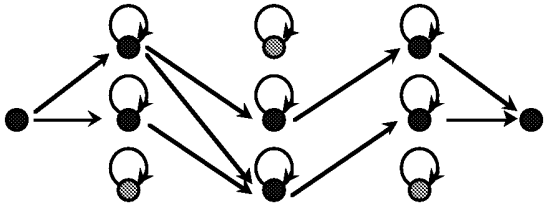


Figure 6. Model of consonant /l/ after reestimation.

Figures 5 and 6 show examples of topology inference on the TIMIT training database with 50-NN pdf. For the vowel /uw/, the topology inference gives a three-states Bakis model, whereas the inference for the consonant /l/ gives a more complex model.

6. CONCLUSIONS

For the first time, the K-NN estimator has been used in a HMM-based system. At the moment, its performances are comparable to the gaussian ones.
Nonetheless, the improvement of the K-NN HMM-based system is expected from the ability to find back and to analyse the training-vectors generating the identification errors. Another possible improvement is the dynamic use of various specific measures. The most sophisticated HMM techniques wil be introduced in the K-NN HMM-based system, such as contextual phonetic models..

REFERENCES

[1] J. Goût, "L'apprentissage en reconnaissance de la parole", Technical Report, Université PARIS 6, 1993.
[2] C. Montacié, M.-J. Caraty & C. Barras, "Mixture Splitting Technic and Temporal Control in a HMM-Based Recognition System", ICSLP, pp. 977-980, 1996.
[3] J. Kittler , "A Method for Determining K-Nearest Neighbours", Kibernetes, vol. 7, 1978.
[4] M. Richetin, G. Rives and M. Naranjo, "Algorithme rapide pour la détermination des k plus proches voisins", RAIRO Informatique, vol. 14, n° 4, 1980.
[5] J.H. Friedman, J.L. Bentley and R.A. Finkel, "An Algorithm for Finding Nearest Neighbours", IEEE Trans. on Computer, July 1975.
[6] K. Fukunaga and P.M. Narendra, "A Branch and Bound Algorithm for Computing K-Nearest Neighbours", IEEE Trans. on Computer, July 1975.
[7] S. Seneff & V. Zue, "Transcription and Alignement of the TIMIT Database", in "Getting Started with the DARPA CD-ROM : An Acoustic-Phonetic Continuous Speech Database", NIST, 1988.
[8] S.J. Young, "HTK Version 1.4 : Reference Manual and User Manual", CUED- Speech Group, 1992.
[9] K.-F. Lee and H.-W. Hon, "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition", IEEE Trans. ASSP, vol. 38, n°4, pp. 599-609, 1990.
[10] F. Casacuberta, E. Vidal, B. Mas, and H. Rulot, "Learning the Structure of HMMs throught Grammatical Inference Techniques", ICASSP, pp. 717-720, 1990.
[11] J. Takami and S. Sagayama, "A Successive State Splitting Algorithm for Efficient Allophone Modeling", ICASSP, pp. 1573-1576, 1992.
[12] R. de Mori, M. Galler and F. Brugnara, "Search and Learning Strategies for Improviong Hidden Markov Models", Computer Speech and Language, vol. 9, pp. 107-121, 1995.