Int. J. Appl. Math. Comput. Sci., 2005, Vol. 15, No. 2, 287–294

ON NAIVE BAYES IN SPEECH RECOGNITION

LÁSZLÓ TÓTH, ANDRÁS KOCSOR, JÁNOS CSIRIK

Research Group on Artificial Intelligence
H–6720 Szeged, Aradi vértanúk tere 1., Hungary
e-mail: {tothl, kocsor, csirik}@inf.u-szeged.hu

The currently dominant speech recognition technology, hidden Markov modeling, has long been criticized for its simplistic assumptions about speech, and especially for the naive Bayes combination rule inherent in it. Many sophisticated alternative models have been suggested over the last decade. These, however, have demonstrated only modest improvements and brought no paradigm shift in technology. The goal of this paper is to examine why HMM performs so well in spite of its incorrect bias due to the naive Bayes assumption. To do this we create an algorithmic framework that allows us to experiment with alternative combination schemes and helps us understand the factors that influence recognition performance. From the findings we argue that the bias peculiar to the naive Bayes rule is not really detrimental to phoneme classification performance. Furthermore, it ensures consistent behavior in outlier modeling, allowing efficient management of insertion and deletion errors.

Keywords: naive Bayes, segment-based speech recognition, hidden Markov model

1. Introduction

Although speech recognition requires the fusion of several information sources, it is rarely viewed as an expert combination problem. Such approaches were abandoned in favor of the hidden Markov modeling technique (HMM) (Huang *et al.*, 2001), which treats speech as a stochastic process. The source of the success of HMM is that it offers a sound mathematical framework along with efficient training and evaluation. The price is that the simplistic mathematical assumptions of the model do not accord with the real behavior of speech. One of these assumptions is the conditional independence of the spectral vectors. Several alternative models have been proposed to alleviate this flaw, but these have brought only modest improvements at the cost of a considerable increase in complexity. Rather than seeking to eliminate the incorrect modeling bias, here we hope to gain a better insight into why HMM performs so well in spite of the unrealistic naive Bayes assumption.

The structure of the paper is as follows: First of all, we define an algorithmic framework that treats speech recognition as a classifier combination problem. It will help us to understand the recognition process from a classifier combination point of view, and also allow us to experiment with alternative combination schemes. After this we show that HMM is just a special case of our algorithm, obtained when applying the naive Bayes rule. We also briefly present a family of alternative technologies, the segmental modeling framework. In Sections 4 and 6 we assess the pros and cons mentioned in the literature regarding the use of naive Bayes in classification. In Section 7 we show that the recognition of speech requires an additional step, namely, the modeling of outliers, and argue that naive Bayes surmounts this obstacle very well. Lastly, in Section 8 we discuss experiments conducted on a small speech corpus to support our assertions, and then provide a summary of the results in Section 9.

2. Speech Recognition as a Classifier Combination Problem

Although speech recognition is obviously a pattern classification task, the most successful solution, hidden Markov modeling, is not a classification algorithm in the strict sense, but a generative model for stochastic random processes. This is because speech recognition does not fit the usual pattern classification framework. That is, most classification algorithms assume that the items to be classified are always represented by the same number of features. In addition to that, both the dimension of the feature space and the number of classes must be reasonably small. In contrast, speech is a continuous stream of acoustic information. Even if we assume that the talker must stop sometimes, the possible utterances vary in length and their number is practically unlimited. A possible solution is to trace the problem back to the recognition of some properly chosen building blocks. During recognition these building blocks have to be found, identified, and the information

they provide needs to be combined. This approach turns speech recognition into a task of classifier combination integrated in a search process.

In the following we present a general speech decoding scheme in the spirit of classifier combination. Firstly, it makes it possible to experiment with alternative combination schemes which could not easily be done within the traditional HMM framework. Secondly, it provides a more intuitive picture of how the whole recognition process works.

Algorithm 1 shows the pseudocode of our generalized speech decoder. Expressed simply, the algorithm works in the following way: Let us assume that our building blocks are denoted by the elements of the symbol set $\mathcal{F}$. Let the speech signal be given by the series of measurements $A = a_1, \ldots, a_T$. The goal of recognition is to map the speech signal A to a series of symbols $F = f_1 \ldots f_n$, where $f_j \in \mathcal{F}$. The algorithm works from left to right, and stores its partial results in a priority queue. Having processed the signal up to a certain point t, the algorithm looks ahead in time and, from the corresponding measurements, it collects evidence that the next symbol belongs to the time interval being inspected. As neither the exact length nor the identity of the next segment is known, we examine every time index $t' = t+1, t+2, \ldots$ that might be the end point of the segment. Each element f of the symbol set is matched to the interval $\langle t, t' \rangle$, and from each (t', f) pair a new hypothesis is formed and put in the hypothesis queue. As every hypothesis has several extensions, this means creating a search tree. By adjusting the hypothesis selection strategy, the pruning and the stopping criteria, one can control how the search space is traversed and pruned.

When the whole signal has been processed, the best scoring leaf is returned as the result. The score of a hypothesis is calculated in two steps. First, there is a function (g_1) to combine the evidences for each symbol as collected from the local information sources. Second, this local evidence is combined (via g_2) with the prefix of the hypothesis to obtain a global score. So, in effect, classifier combination occurs at two levels.

Obviously, we obtain quite different decoders depending on how the measurements a_i, the symbol set $\mathcal{F}$ and the functions g_1 and g_2 are chosen. Researchers agree only in that g_1 and g_2 should work on probabilistic grounds. In this case Bayes' decision theorem guarantees optimal performance, and statistical pattern recognition provides methods for approximating the probabilities from training corpora.

The acoustic information sources a_i display the greatest variation from system to system. Traditionally, the acoustic signal A is processed in small uniform-sized (20–50 ms) chunks called "frames", and the spectral representation of these serves as a direct input for the model. It has been observed, however, that better results are obtained if this representation is augmented with features of longer time-spans so the feature vectors in current systems are a combination of the local and the neighboring 5–50 frames (Huang *et al.*, 2001).

As regards the selection of the building units, the most reasonable choice is the phoneme, since phonemes are the smallest pieces of information carrying units of speech (in the sense that the insertion/deletion/substitution of a phoneme can turn a word into another one). Furthermore, in many languages there is an almost a one-to-one correspondence between phonemes and letters, so working with phonemes is an obvious choice when converting sound to writing. Nevertheless, smaller or larger units could be used as well. For example, there are arguments that syllables give a more suitable representation of (the

Algorithm 1. Generalized Speech Decoding Algorithm

```
solutions := ∅
hypothesis queue := h_0(t_0, "", 0)
// a hypothesis consists of a time index, a phoneme string, and
// a score
while there is an extendible hypothesis do
  select an extendible hypothesis H(t, F, w) according to
  some strategy
  if t = T then
    if only the first solution is required then
      return H
    else
      put H on the list of solutions
    end if
  end if
  for t' = t+1, t+2, ··· do
    for all f ∈ F do
      w_f := g_1(f, ⟨t, t'⟩) // where g_1 estimates the cost of
      fitting f to ⟨t, t'⟩
                              // based on the relevant a_i mea-
      surements
      w' := g_2(w, w_f)      // where g_2 calculates the cost of
      attaching f to the
                              // hypothesis prefix F
      if pruning-criterion (w_f, w') then
        construct a new hypothesis H'(t', Ff, w') and put it
        in the hypothesis queue
      end if
    end for
    if stopping-criterion(⟨t, t'⟩) then
      break
    end for
  end for
end while
```

English) language. Going the other way, current recognizers mostly decompose phonemes into three articulation phases (Huang *et al.*, 2001).

Linguistic information, for example, phone or word N-grams, pronunciation dictionaries or formal grammars can be incorporated into the recognition process via g_2, as this component is responsible for concatenating the building units into a string of symbols. Probabilistic language models will take the form of multiplying factors, while formal grammars appear as constraints that reject certain unit combinations.

3. Special Case: Hidden Markov Models

In spite of its unusual appearance, *Algorithm 1* is not so different from the standard technologies. In particular, its components can be chosen so that it becomes mathematically equivalent to phoneme-based left-to-right hidden Markov modeling preferred in large vocabulary speech recognition. In this setup the set of states of the Markov model will play the role of the symbol set in our algorithm.

Instead of modeling the class posteriors $P(F|A)$ directly, in speech recognition the product $P(A|F)P(F)$ is normally modelled instead, which leads to the same result but allows one to separate the priors $P(F)$. Building words from states and assessing their prior probability $P(F)$ is the problem of language modeling. So let us first deal only with the acoustic component $P(A|F)$. This factor will be estimated by HMM in the way described below[1].

During operation HMM goes through a sequence of state transitions. This determines a segmentation based on how long the model stayed in a given state. The probability corresponding to a given segmentation is calculated as follows: The probability corresponding to a given segment $S_i = \langle t, t' \rangle$ and state f is calculated as

$$P(\langle t, t' \rangle | f) = l_f^{(t'-t)} \prod_{i=t}^{t'} P(a_i|f), \tag{1}$$

where l_f is a constant between 0 and 1.

The probability corresponding to the whole segmentation is obtained by multiplying the segmental probabilities:

$$P(A, S|F) = \prod_{i=1}^{n} P(S_i|f_i). \tag{2}$$

[1] Note that we slightly deviate from the standard decomposition into language and acoustic models as, in our notation, the state transitions between the states of a multi-state acoustic model are also included in the language factor, while only the self-transitions of a state are included in the acoustic model.

As the last step, the product $P(A|F)P(F)$ is obtained by multiplying $P(A, S|F)$ with the language model factor $P(F)$ and removing S by searching and maximizing over all possible segmentations during the recognition process of *Algorithm 1*.

Let us assume for a moment that no linguistic information is available, i.e., $P(F)$ is the same for any symbol string, and thus this component plays no role during decoding. Then, in terms of our model, Eqn. (1) corresponds to g_1 while Eqn. (2) corresponds to g_2. This means that g_2 is simply a multiplication, while g_1 consists of two factors. The term $l_f^{(t'-t)}$ is an exponentially decaying duration model. The product $\prod_{i=t}^{t'} P(a_i|f)$ is a spectral factor that renders a state-conditional likelihood for each measurement of the segment, and then combines these by multiplication—that is, by applying the naive Bayes assumption. This factor is the focus of the paper that we intend to examine in greater detail.

As we wanted to concentrate on the acoustic component in the experiments, we chose the simplest possible setup where the states simply represent phonemes and $P(F)$ is either a unigram model that permits any possible phoneme string, or a pronunciation dictionary that simply restricts the accepted phoneme strings to a small set. If we were using a more stochastic language model, its scores should be incorporated into the evaluation of g_2. Moreover, one may ask what would happen if one were to work with acoustic units other than simple 1-state phoneme models, as we do here. Clearly, in practice, better results are normally obtained if the phonemes are decomposed into three states—one corresponding to the middle steady-state part, and the others describing the transitional phase before and after. If we were to use such a three-state model, then multiplication by the interstate transition probabilities should be incorporated into g_2. Improving the model further by applying context-dependent models such as diphones or triphones would correspond to a refinement of the symbols set and, naturally, the associated phonetic transcripts or other language components. Although all these modifications could improve the performance of the system, they all affect g_2 and not the acoustic component g_1 we are dealing with here. Thus, all our arguments regarding the naive Bayes assumption remain valid irrespective of the symbol set and the language model used, as far as the acoustic information sources are frame-based likelihoods combined by multiplication.

4. Naive Bayes: the Cons

The hidden Markov technique is a general mathematical framework for modeling stochastic sequences. Its main

power is its mathematical tractability—that it can be evaluated very quickly by dynamic programming, and that its (locally) optimal parameters can be found relatively simply (Huang *et al.*, 2001). However, whether these optimal parameters provide a good performance also depends on how well the modeling assumptions fit the given application. HMM has a very serious inductive bias as it assumes the state-conditional independence of the acoustic vectors. In contrast, the neighboring frames are obviously correlated as speech is produced by a continuous movement of the articulators. Moreover, many signal processing methods applied in the feature extraction step increase the correlation as they linearly combine the neighboring data vectors. As a coup de grâce, we extend out feature set with the so-called delta features, which are again obtained as a combination of a few neighboring frames (Huang *et al.*, 2001).

Based on speech perception experiments, we can also argue against combination by multiplication. Namely, it is known that humans can recognize speech quite well even when large portions of the spectral information are removed. In comparison, the production combination rule is too restrictive in the sense that any frame can 'veto' the classification by making the product zero.

As a final argument, the classifier combination literature suggests that, in general, the production rule performs well when the classifiers work on independent features. When the features contain similar information—as in our case—then other schemes like combination by averaging are likely to yield better classification results (Tax *et al.*, 2000).

5. Alternative Technology: Segmental Models

The contradiction between the model that assumes independence and the feature extraction method that makes it patently false has been understood and criticized by many authors (Ostendorf *et al.*, 1996; Van Horn, 2001). Several cures were suggested, some only patching the original HMM algorithm, some totally abandoning it. The family of segmental models (Ostendorf *et al.*, 1996) recommends modeling phonemes 'in one', instead of estimating their probabilities by multiplying the frame-based scores. In our framework this means that g_1 (Eqn. (1)) is replaced by some more sophisticated approximation[2]. One possibility might be to create special models that, for example, fit parametric curves on the feature trajectories (Holmes and Russel, 1999; Ostendorf *et al.*, 1996). Another option is to convert the variable-length segmental data into a fixed number of segmental features (Clarkson and Moreno, 1999; Glass, 1996; Tóth *et al.*, 2000). What makes the latter tempting is that this way all the standard classification algorithms become applicable to the phoneme classification task.

Whatever technique we choose, the results are similar. Searching in the literature we find that these models result in a 10–30% reduction in the phoneme classification error compared to HMM (see, e.g., (Clarkson and Moreno, 1999; Holmes and Russel, 1999)). Although this is significant, it is rather modest considering that we have replaced an incorrectly biased model with a much better one.

6. Naive Bayes: the Pros

Many have critized the use of the naive Bayes assumption in HMM. But we are unaware of anyone in the speech community putting the question the other way round: why does it work so remarkably well when, in theory, it should not? Fortunately, we can find partial answers in the machine learning literature. Most pertinently, it has been pointed out that in many cases naive Bayes provides optimal classification even though it incorrectly estimates the probabilities (Domingos and Pazzani, 1997). One such case is when there is full functional dependency between the features (Rish *et al.*, 2001). Even when the dependency is not completely deterministic, the naive Bayes classification was found to perform nearly optimally in (Rish *et al.*, 2001). The explanation is that in these cases all features yield approximately the same probability estimates, so when we combine them by multiplication it is like raising one output to the number of classifiers combined. The resulting estimation tends to underestimate the real probabilities. Besides this, the probability value of the winning class dominates over that of the others. Quoting Hand, "the model will have a tendency to be too confident in its predictions and will tend to produce modes at the extremes 0 and 1" (Hand and Yu, 2001). However, these values lead to the same classification as raising the estimates to a power preserves the rank order.

Knowing that the feature vectors in speech recognition are highly correlated, we might suspect that a similar effect must occur with HMMs. It has indeed been reported that HMMs are "overconfident of their recognition results" (Van Horn, 2001), and that "primarily due to invalid modelling assumptions, the HMM underestimates the probability of acoustic vector sequences" (Woodland and Povey, 2000). This supports our argument and taken together may explain why HMMs perform well in phoneme classification in spite of the manifestly false independence assumption.

[2] In contrast to g_1, combination by multiplication at the level of g_2 seems quite reasonable because the presence of all phonemes is required for the identity of a word. This makes an AND-like combination logical.

7. From Classification to Recognition

Thus far we seem to have overlooked the fact that, as part of the recognition system, the role of phoneme models is not classification, but rather probability estimation! At first sight this seems to invalidate all our arguments for naive Bayes, making the explanation of its efficient classification irrelevant. In the following we are going to argue that during recognition the phoneme classification task is simply extended with outlier modeling, and the bias of naive Bayes is not harmful to the latter step either.

First of all, let us clarify what happens when we move from classification to recognition. During classification we assume that the start and end points of the phonemes—that is, the correct segmentation of the signal—were known. Consequently, the only task was to identify the segments. During recognition, however, the proper segmentation also has to be found. This requires discriminating real phoneme segments from fake ones. Note that we neither have a model dedicated to these non-phonemic segments, nor training examples for them. This means that we are faced with an outlier modeling problem. If our phoneme model is not able to reject these outliers, it will then be prone to commit insertion and deletion errors. That is, it is going to cut the phonemes into more segments or fuse the frames of a segment with neighboring segments.

Let us now examine how the hidden Markov model behaves when it is allowed to evaluate all state sequences and segmentations. Obviously, the model gives the highest value if the signal is cut into 1-frame long segments, and for each of these the state with the highest likelihood is selected. This is avoided by the language model that punishes unlikely state transitions and/or excludes impossible ones. In this way we can force the model to fuse neighboring frames, but even in this case it will have a strong preference for short segments. This is because the frame-based likelihoods are very small (non-negative) values, so when we multiply them we will get progressively smaller values for progressively longer segments. Another factor is, of course, the exponentially decaying duration component. However, because the spectral likelihoods are usually many orders of magnitudes smaller than l_f, it has been reported by many researchers that it has virtually no effect on the recognition performance. This means that it is practically the naive Bayes combination rule that drives the system towards short segments.

When forced to fuse neighboring frames, the model will prefer those subsegments in which one of the states provides consistently high values. It is fulfilled if the system performs reasonably well at the frame level. It is also known that the frame-level classification tends to be more stable in the middle of the segments and more inconsistent at the segment boundaries. This will 'push' the model towards fusing the central parts of the phonemes and position the state transitions near the real segment boundaries.

8. Experiments

To justify our conclusions we conducted experiments to assess the influence of naive Bayes on both classification and recognition performance. For this purpose we replaced the naive Bayes product rule with alternative combination formulas. Comparing these results gave an indication of how beneficial or detrimental naive Bayes is on classification and on outlier modeling.

In the experiments the "Oasis-Numbers" speech corpus was used. Its data were collected at our institute and consist of spoken numbers, recorded with several types of microphones at a sampling rate of 22050 Hz in the 16-bit quality. The whole corpus is manually segmented and labeled at the phoneme level. Altogether 29 different phonemic labels occur in the transcripts. 2185 and 1247 utterances were randomly selected for training and testing purposes, respectively.

For feature extraction we utilized the HCopy routine of the HTK toolkit (Young *et al.*, 2004). We extracted 13 MFCC coefficients from each frame, along with their first and second derivatives. This feature set is the most widely used one in speech recognition (Huang *et al.*, 2001).

Modeling whole segments in one requires an additional step. The variable-length frame-based representation has to be converted into a fixed-dimensional feature set. To achieve this we used the simple method proposed in the SUMMIT system (Glass, 1996), but also successfully applied by us (Tóth *et al.*, 2000) and others (Clarkson and Moreno, 1999). The segments were divided into three parts along the time axis, and each frame-based feature was averaged over these thirds. Additionally, the length of the segment was also included in the segmental feature set.

For modeling the frame-level and segmental likelihoods, Gaussian mixtures were applied, which is again a standard technology in speech recognition. The model parameters were initialized by K-means clustering and trained with Expectation Maximization. 15 Gaussian components performed the best in the frame-level and 10 in the segmental modeling task. In both cases the covariance matrices were kept diagonal.

8.1. Classification

In the classification experiments we utilized the manual segmentation information of the database. This means that the search part of our decoding algorithm was deactivated by restricting the decoder to evaluate only the

Table 1. Classification and recognition accuracies

Phoneme model	Classification accuracy	Recognition acc.	
		Unigram	Vocabulary
Frame-based, product rule	92.33%	82.05%	96.87%
Frame-based, averaging rule	78.04%	—	86.28%
Frame-based, product rule, n-th root	92.33%	—	41.78%
Segmental	94.58%	46.25%	87.00%
Segmental, n-th power	94.58%	57.99%	88.29%

correct segmentation. All phoneme priors were assigned equal values in these experiments.

The percentage of correctly classified segments is shown in the first column of Table 1. Besides the segmental representation and the one that combines the frame-level likelihoods by multiplication, out of curiosity we also tried combination by averaging. Furthermore, we tested two further possibilities. The first one was to compensate for the bias of the product rule by taking the n-th root of its likelihood estimations, where n is the number of frame-based scores multiplied (as suggested in (Hand and Yu, 2001)). The other idea was to introduce a similar bias into the segmental model by raising its estimates to the n-th power. These manipulations clearly do not influence classification. However, they result in quite different likelihood estimates that may seriously affect the search process.

We have to emphasize that our goal here was not to achieve high-performance classification but to compare the two approaches. The product rule combination of the frame-based likelihoods corresponds to a 1-state hidden Markov model, which could be outperformed by the usual 3-state representation. The segmental model could also be improved by adding further features. The results nevertheless reflect quite well the usual findings when comparing segmental models with HMMs, which is the modest superiority of the segmental representation.

We did not mention that the frame-based Gaussian models were able to classify 71.54% of the frames correctly. The product rule brought substantial improvement compared to this, while averaging outperformed it only modestly. A possible explanation is that when a frame is classified correctly, the likelihood of the correct class is much higher than those of the competing ones. And if a frame is misclassified, the likelihood of the correct class is still relatively high. As a consequence, the product rule does not get fooled by the erroneous frames, but the dominance of the correct ones tilts the product in the right direction. Averaging profits less from the high confidence of the correct decisions, and so is more vulnerable to the incorrect ones.

8.2. Recognition

In the recognition experiments the algorithm was allowed to evaluate every possible segmentation. The segmental probabilities were calculated exactly as described in the previous section, but now as a part of the whole search process.

As regards language modeling—that is, the prior probabilities of phoneme sequences—two extreme cases were tried. In one case every phoneme was allowed to follow a phoneme, and with equal probability. This could be called a 'unigram' language model. In the other case the possible phoneme sequences were restricted to a 26-word vocabulary, each word being equally probable. This corresponds to a very small vocabulary isolated word recognition task.

The scores reported when using the dictionary are simply the percentage of words recognized correctly. In the case of the unigram model, however, the result of recognition is a phoneme sequence that, besides misclassifications, can contain insertion and deletion errors as well. The standard evaluation method is to match the result with the manual phonemic transcription by calculating their edit distance (Young *et al.*, 2004). Having obtained the best match, all three types of error are counted and included in the accuracy score.

When testing the product rule with the unigram model we found that—in accordance with our expectations—insertion errors tended to overwhelm the result. We compensated for this by raising the language model probabilities to an empirically tuned factor. Following (Lee and Hon, 1989), this factor was adjusted so that the insertion errors went down to about 10% of the number of test instances. A similar language model compensation was applied in every case when insertion or deletion errors became seriously unbalanced.

The results are listed in the last two columns of Table 1. The most important finding is that the frame-based model with the product rule performed the best with both language models, and the segmental model could not even come close. This shows that better phoneme classification does not automatically warrant better recognition. Because the segmental model is not designed to refuse outliers, a segmental recognizer needs a further component to handle them. Although with such modifications the segmental technology may outperform HMM, this means a further algorithmic and computational burden compared to HMM that 'automagically' handles this problem.

A further observation was that the product rule displayed very consistent behavior regarding insertion and deletion errors. This means that by adjusting the weight of

the language model we could easily tune the ratio of insertions and deletions in the Unigram experiments. In comparison, with the averaging rule we were unable to obtain reasonable results because certain phonemes tended to 'eat up' their neighbors, while some others were cut into lots of small segments. The segmental model displayed quite similar rhapsodic behavior, although to a lesser extent. Besides insufficient outlier modeling, weak duration modeling may also contribute to this. Although the segmental duration was among the features and, in theory, the model had the option of making use of it, we noticed that the model still allowed ridiculously long or short segments.

As regards the compensation experiment, taking the n-th root had a fatal result on recognition, leading to the chaotic behavior just mentioned. However, we have probably overcompensated for the bias of the product rule, so the experiment where we introduced a similar bias into the segmental model might be expected to yield more conclusive results. It showed that raising to a power did not cause any harm. Actually, it led to a slight improvement. It indicates that an incorrect bias that severely punishes long segments performs better in finding the correct segmentation than a model that has no idea of fake segments and is not really good at duration modeling anyway.

Finally, we should also mention that segmental models are more prone to variance problems due to insufficient data, as of course there are many more frames than phonemes. This may also contribute to the instability of the segmental system.

9. Conclusions

This paper sought to gain an insight into why HMM speech recognizers, built on the naive Bayes assumption, perform so well. We argued that speech recognition consists of two subtasks, namely, phoneme classification and outlier modeling, and that the naive Bayes rule does well in both tasks. As regards classification, we have pointed out that the data frames are not independent, but in fact just the opposite: they are highly correlated. However, we found evidence from the literature that this condition, although being detrimental on the resulting probability estimates, does not necessarily lead to poor classification. But this still does not explain why the recognition process is not fooled by the naive Bayes assumption, since during recognition the probability estimates are used, and not simply the classification results. We explained this here by pointing out that the probability estimates of the naive Bayes rule are such that they get smaller and smaller for longer and longer segments. This biases the model towards a strong preference for short segments, especially where the probability of one class is consistently high. This was clearly justified by the fact that in practice, when only a phone-unigram was used, a phone insertion penalty term had to be introduced, otherwise insertion errors overwhelmed the result. However, by carefully tuning this parameter or by using a pronunciation dictionary, this bias of the model could be nicely counterbalanced, so altogether we can say that naive Bayes manifests itself in consistent and nicely manageable behavior from an outlier modeling point of view.

To underpin our arguments, a small set of experiments were also carried out where we compared the product rule with a segmental representation. We found that the segmental model performed only slightly better in classification and, in spite of acting better as a classifier, provided much worse recognition. Overall this shows that the simple product rule, although suboptimal, warrants stable and reliable behavior along with a decent recognition performance. In comparison, segmental recognizers have to take more care of outliers in order to obtain similar or better recognition results. Although the phoneme models themselves could also be improved to reject fake segments, it is probably more effective to model them explicitly. This can be done by introducing an 'anti-phoneme' model (Glass, 1996; Tóth *et al.*, 2000) or by assessing probabilities for the different segmentations and incorporating this factor into the formulas (Verhasselt *et al.*, 1998). Although with such extensions the segmental representation is known to be able to produce a modestly better performance than the traditional HMM models, the complications and inconveniences introduced by the need for such a factor makes the segmental modeling paradigm even less attractive. We hope that our arguments and experiments helped to shed light on why HMMs—built on the very simple naive Bayes assumption—behave so well in practice that quite complex alternative models like the segmental model can hardly compete with them.

References

Clarkson P. and Moreno P.J. (1999): *On the use of support vector machines for phonetic classification.* — Proc. IEEE Int. Conf. *Acoustics, Speech and Signal Processing*, Phoenix, AZ, pp. 585–588.

Domingos P. and Pazzani M. (1997): *On the optimality of the simple Bayesian classifier under zero-one loss.* — Machine Learn., Vol. 29, No. 2–3, pp. 103-130.

Glass J.R. (1996): *A probabilistic framework for feature-based speech recognition.* — Proc. 4-th Int. Conf. *Spoken Language Processing*, Philadelphia, PA, pp. 2277–2280.

Hand D.J. and Yu K. (2001): *Idiot's Bayes—Not so stupid after all?* — Int. Stat. Rev., Vol. 69, No. 3, pp. 385–398.

Holmes W.J. and Russel M.J. (1999): *Probabilistic-trajectory Segmental HMMs*, — Comput. Speech Lang., Vol. 13, No. 1, pp. 3–37.

Huang X.D., Acero A. and Hon H-W. (2001): *Spoken Language Processing.* — New York: Prentice Hall.

Lee K.-F. and Hon H.-W. (1989): *Speaker-independent phone recognition using hidden Markov models.* — IEEE Trans. Acoust. Speech Signal Process., Vol. 37, No. 11, pp. 1641–1648.

Ostendorf M., Digalakis V. and Kimball O.A. (1996): *From HMMs to segment models: A unified view of stochastic modeling for speech recognition.* — IEEE Trans. Acoust. Speech Signal Process., Vol. 4, No. 5, pp. 360–378.

Tóth L., Kocsor A. and Kovács K. (2000): *A discriminative segmental speech model and its application to hungarian number recognition.* — Proc. 3rd Workshop *Text, Speech, Dialogue*, Brno, Czech Republic, pp. 307–313.

Rish I., Hellerstein J. and Thathachar J. (2000): *An analysis of data characteristics that affect naive Bayes performance.* — IBM Technical Report RC1993.

Tax D.M.J., van Breukelen M., Duin R.P.W. and Kittler J. (2000): *Combining multiple classifiers by averaging or by multiplying?*. — Pattern Recogn., Vol. 33, No. 9, pp. 1475–1485.

Van Horn K.S. (2001): *A maximum-entropy solution to the frame-dependency problem in speech recognition.* — Tech. Rep., Dept. Computer Science, North Dakota State Univ.

Verhasselt J., Illina I., Martens J.-P., Gong Y., Haton J.-P. (1998): *Assessing the importance of the segmentation probability in segment-based speech recognition.* — Speech Commun., Vol. 24, No. 1, pp. 51–72.

Woodland P.C. and Povey D. (2000): *Large scale discriminative training for speech recognition*. — Proc. ISCA ITRW ASR 2000, France: Paris, pp. 7–16.

Young S. et al. (2004): *The HMM Toolkit (HTK) (software and manual)*. — Available at `http://htk.eng.cam.ac.uk/`

Received: 30 July 2004
Revised: 21 March 2005