An HMM-Based Approach to the INTERSPEECH 2011 Speaker State Challenge

Albino Nogueiras Rodrigues

TALP Research Center. Universitat Politècnica de Catalunya. Barcelona, Spain. albino.nogueiras@UPC.edu

Abstract

The current main trend in paralinguistic information recognition is the so-called static classification. In this kind of classification the low level descriptors are pooled together by means of statistical functionals and all, or almost all, information about the temporal structure and evolution of speech is lost. Although this approach represents the state-of-the-art, we believe that dynamic classification, where temporal information is kept, still deserves some attention due to its capability to handle aspects impossible to do by the static one.

In this paper the INTERSPEECH 2011 Speaker State Challenged is addressed using the Automatic Speech Recognition system developed at UPC, which has already been used in a similar task: emotion recognition. Although results fall below the baseline, we believe that they are close enough to be taken into account.

Index Terms: Speaker State Challenge, paralinguistics, Hidden Markov Models.

1. Introduction

Over the last few years, interest on paralinguistic information classification has grown considerably. The successive challenges proposed at the last INTERSPEECH conferences are a clear evidence of this. As these challenges show, there are countless tasks that can be taken into account, being emotion, gender and age just some examples of the variety and variability of them. This variability implies that each classification task needs specific information and tools to be performed. Nevertheless, they all share -apart from, perhaps, emotion recognition– one common feature: they do not depend on time. Thus, it is not predictable that the emotional state of the speaker varies much during the utterance. Needless to say gender or age. This is also the case in the current challenge: one can expect the speaker to keep his state of alcoholic intoxication and sleepiness during the whole utterance.

The fact that the paralinguistic information is not expected to change during the utterance enables us to undertake the problem using it as a whole: large vectors of low level descriptors are estimated, then even larger vectors of statistical measures of these descriptors are constructed, finally these vectors are used by different kinds of linear or non-linear classifiers to provide the result. Somehow, as more information is passed to the classifier, the better the result is. And this means increasingly big vectors. It is what has been called brute-force extraction, which is clearly exemplified in the challenge paper [1]. There, three sets of features are used to provide the baseline results: the first, with only 384 features, leads to the worst result; the second, with 1582 features, clearly outperforms the former; and the third, with 4368 features, is the best of them all.

As already said, the abovementioned standard classification approach is based on the assumption that the paralinguistic information to be detected does not change over time. Yet, the used low level descriptors are taken from voice signals, and voice does change over time. A clear implication of this is that the statistics taken over the whole utterance will depend on its phonetic contents. This is not much of an issue when the utterance is long: we can expect to find similar distributions of phones on different long sentences. But it may become a problem for shorter utterances, where the phonetic contents may vary a lot from one sentence to another.

In our proposal, we try to cancel the effects of the phonetic contents of the utterance by reflecting this variability in the state structure of large hidden Markov models (HMM). To do so, we build a different HMM for each speaker state to be recognized in such a way that the sequence of states followed along the HMM does not depend on the speaker state itself but on the phonetic evolution of the sentence, and is the same for all the speaker states considered. This approach had already been used in a similar task, emotion recognition [2], leading to good results, comparable to those achieved in the same task using global statistics [3] [4].

The structure of the paper is as follows: Section 2 overviews the system we use to address the challenge; Section 3 shows the results we obtain; and Section 4 makes some conclusions and draws some future work lines.

2. System overview

2.1. Feature extraction and rank reduction

The basic features used in this paper are the set of low level descriptors (LLD) provided by the organizers. We did not use any of the functionals proposed, but modified the scripts in order to get the 60 LLDs every 10ms.

In order to capture the dynamics of speech, 11 frames of the basic features and their utterance mean are joined to form a 720-dimension vector that is subsequently reduced to 64 components using linear discriminant analysis (LDA).

2.1.1. Linear discriminant analysis

Contrary to classifiers based on whole utterances, where an increase in the dimension of the feature space seems to lead to better results, HMM modeling requires that this dimension is kept relatively small. In order to do so, a popular choice is LDA. For a given classification task in C different classes, and in the case that the feature vectors belonging to each of the classes are normally distributed with equal covariance matrices, LDA provides the best linear classifier in terms of classification error. This is accomplished by finding the directions in the feature space that provide the maximum separability, i.e. the ratio between the inter-class separation and the intra-class variability (Fisher's linear discriminant).
This linear classifier may be directly used, but is usually not because there are other more powerful tools than it. Yet, its ability to find the directions with the maximum separability is still useful for reducing the dimension of the feature vectors by keeping these components, supposed to carry most of the information needed for the classification task, while discarding the rest, supposed to carry mainly noise.

The problem in using LDA for classification tasks such as those involved in this challenge is that it just can find C-1 directions in the feature space, where C is the number of different classes. This means that, in the intoxication and sleepiness challenges, just one component would be kept in the feature vector –two if silence is also considered a different class–. And this is clearly not enough.

In our work, we use a rather different alternative: instead of trying to optimize the discrimination in the original classification task, we optimize it using as classifier the vector quantifier provided by a similar codebook to that used by the HMMs. As will be explained later, the algorithm we use to train codebooks and HMMs tries to place the codewords in positions where the discrimination of the models is maximized. Thus, the distribution of the codewords in the feature space is a good representation of how useful this space is in discriminating among the different classes. The procedure is as follows:

- Initially, we discard from the LLD set the 26 RASTA-style spectrum energies because they increase much the dimension of the feature space without providing substantial additional information.
- The resulting 34 coefficients vector is increased with its first and second deltas, resulting in a 102-dimension feature space.
- HMMs and codebook are trained for this LLD set.
- A new feature set is constructed using the 60 original LLDs in windows of 11 frames plus the utterance mean vector.
- Using as classifier the codebook previously trained, the LDA matrix is estimated for this expanded feature set.
- The 720-dimension expanded feature set is reduced to its 64 most relevant components using the LDA matrix.

### 2.2. Acoustic modeling

For modeling the different speaker states we use our standard ASR system based on semi-continuous HMMs. Although this kind of modeling is often deprecated in front of continuous HMMs, our experience in ASR shows that differences are not so big and, moreover, not always in favor of the later. It is more a question of how models are trained, than its structure.

In our case we use an algorithm that jointly trains models and codebooks, leading to a solution that is very close to fully tied continuous HMMs [5]. It consists in starting with a codebook of size one and grow simultaneously the codebook and the HMMs until the desired size is reached. At every size in the growth, the HMMs and the codebook are re-estimated using expectation maximization. In order to grow the codebook, LDA is performed for every codeword and the direction of maximum discrimination is used to split it into two.

In this way, we get a codebook where the different codewords are equivalent to the Gaussian distributions used in fully-tied continuous HMMs. The main difference with more standard continuous HMMs is that all the Gaussians are used in all the states. The advantages of this modeling in front of continuous HMMs come from the fact that the feature space is actually quantified: at every point in the space the active Gaussians are the same for all the states of all the models. Yet, this quantization is very different to what could be expected in traditional semi-continuous modeling where the codebook is trained first with k-means. For instance, we have no problem in dealing with feature spaces of dimension 100 or more.

#### 2.2.1. Codebook training

In the scope of the present work, this way of training codebooks and models presented one problem: the LDA used to grow the codebook uses the states of the HMMs as the classes to be discriminated, but there are only two speaker states we are willing to distinguish, and the HMM of each of them has 32 states. This means that the codebook would be much more specialized in distinguishing the states of each HMM, than in the speaker state. To avoid this problem we divided the training in two phases: in the first we jointly train a codebook of 2000 codewords with 1-state HMMs; then we quantify the utterances, drop the 1-state models and build the definitive 32-states models with standard semi-continuous training.

#### 2.2.2. HMMs structure

The structure of the HMMs plays a fundamental role in our system. We want the HMMs to model voice in utterances of arbitrary length. This means that each HMM should have a number of states enough to model the different sounds that can occur. Moreover, these sounds can occur in arbitrary order, so loops should be possible. In the structure that we finally use each HMM has 32 states, being possible entries to the model the first two states and possible exits the last two. All backwards transitions are allowed. Besides, and in order to ensure that only long enough segments are assigned, we limit the number of states that can be skipped in the forwards transitions to three. This implies that at least 11 frames must be spent in the model, with all the states accessible in a few frames.

#### 2.2.3. HMMs training

The main idea in our proposal is that both speaker states are voice, identical except in the cues that mark each of the speaker states. This means that both models must be very similar in the sense that we only want them to differ in these cues. Whichever other peculiarities of the voices training them other than the speaker state must be shared. In order to ensure this, we first use the Baum-Welch algorithm to train a Universal Background Model, similar to those used in speaker identification [6]. This model is meant to characterize the different sounds that appear in voice. This strategy is equivalent to the used in [7] for the 2010 INTERSPEECH Paralinguistic Challenge, but applied to HMMs instead of Gaussian Mixture Models.

Once we have the common voice model, it is used to initialize the HMMs of each speaker state, and we re-estimate them using discriminative training. As the training material is highly unbalanced, while the objective score is the un-weighted recall, we decided to use Minimum Confusability training with equal relevance given to each of the speaker states [8].

#### 2.3. Language modeling

Obviously, the language model for these two tasks is very simple, as only two classes are to be distinguished. Yet, two different strategies can be applied to the treatment of silence.
Table 1. Intoxication and Sleepiness Sub-Challenges results by unweighted and weighted accuracy (UA/WA). Results obtained by the baseline using the feature sets of the INTEESPEECH Challenges 2009, 2010 and 2011, are given for comparison purposes. UPC system uses the IS2011 SCC feature set.

<table>
<thead>
<tr>
<th>Intoxication Sub-Challenge</th>
<th>System</th>
<th>%UA</th>
<th>%WA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train vs. Develop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS 2009 EC</td>
<td>57.4</td>
<td>65.3</td>
<td></td>
</tr>
<tr>
<td>IS 2010 PC</td>
<td>61.6</td>
<td>66.1</td>
<td></td>
</tr>
<tr>
<td>IS 2011 SCC</td>
<td>65.3</td>
<td>69.2</td>
<td></td>
</tr>
<tr>
<td>UPC</td>
<td>63.0</td>
<td>65.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Train+Develop vs. Test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS 2009 EC</td>
<td>60.3</td>
<td>60.2</td>
<td></td>
</tr>
<tr>
<td>IS 2010 PC</td>
<td>63.2</td>
<td>62.6</td>
<td></td>
</tr>
<tr>
<td>IS 2011 SCC</td>
<td>65.9</td>
<td>66.4</td>
<td></td>
</tr>
<tr>
<td>UPC</td>
<td>60.1</td>
<td>60.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sleepiness Sub-Challenge</th>
<th>System</th>
<th>%UA</th>
<th>%WA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train vs. Develop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS 2009 EC</td>
<td>65.3</td>
<td>64.2</td>
<td></td>
</tr>
<tr>
<td>IS 2010 PC</td>
<td>65.1</td>
<td>66.4</td>
<td></td>
</tr>
<tr>
<td>IS 2011 SCC</td>
<td>67.3</td>
<td>69.1</td>
<td></td>
</tr>
<tr>
<td>UPC</td>
<td>64.8</td>
<td>66.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Train+Develop vs. Test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS 2009 EC</td>
<td>68.0</td>
<td>72.4</td>
<td></td>
</tr>
<tr>
<td>IS 2010 PC</td>
<td>70.2</td>
<td>72.8</td>
<td></td>
</tr>
<tr>
<td>IS 2011 SCC</td>
<td>70.3</td>
<td>73.0</td>
<td></td>
</tr>
<tr>
<td>UPC</td>
<td>66.3</td>
<td>68.4</td>
<td></td>
</tr>
</tbody>
</table>

In the first, explicitly modeled silence can occur at any point in the utterance. Thus, each speaker state is modeled with any combination of its model and that of silence. In the second, silence is only explicitly modeled if it occurs at the beginning or the end of the utterance, embedding internal silences in the model of the speaker state.

Performance differences between these two strategies are small, but consistently favor the second, probably because embedding internal silences in the speaker state HMM force it to take into account the frequency and duration of these silences.

3. Experimentation

3.1.1. Intoxication Sub-Challenge

The described system was initially used in the Intoxication Sub-Challenge. Experimental results can be found on the left of Table 1. As it can be seen, they clearly fall below the results obtained by this year's baseline. Nevertheless, they are comparable to those achieved with the baseline when it uses the previous years' feature sets. This could reflect that the LLDs and functionals added to this year's official set are specially suited for its use with systems based on Support Vector Machines, or similar. As a matter of fact, the addition of the RASTA-style filtered auditory spectrum did not represent any improvement in our results, and we used none of the functionals, present or past.

Further, it is also remarkable the fact that our results are comparatively much better when evaluating the development corpus with the train corpus, that when evaluating the test corpus with the combination of the train and development corpora. In the former case, our system clearly outperforms the baseline one using the IS 2009 EC feature set, and is even better than the baseline when it uses the IS 2010 PC features. Nevertheless, when the comparison is done with the test corpus, our system only equals the results of the baseline using the IS 2009 EC feature set. This discrepancy could be due to an excessive adaptation of the UPC system parameters to the recognition of the development corpus.

It must be also noted that close analysis of the recognition results showed that many errors were due to the fact that the assumption that the same path was followed along the two speaker states HMMs was not accomplished. We are currently working in algorithms to solve this problem.

3.1.2. Sleepiness Sub-Challenge

The same system as presented to the Intoxication Sub-Challenge was tried on the Sleepiness Sub-Challenge. Initially, several simplifications were done due to lack of time. For instance, we did not perform a specific LDA estimation for this task, using instead the features estimated for the Intoxication Sub-Challenge. The results are shown on the right of Table 1. In this case, our system only equals the baseline system using the IS 2009 EC and IS 2010 PC feature sets when we evaluate the classification of the development corpus using models trained with the train corpus, and performs worse than all three when the evaluation is carried on the test corpus.

We expected that better results would be achieved when a more specific system could be developed, but when we did it no noticeable improvement was found on the Train vs. Development experiment. Probably, the bad results here are due to the nature of the task itself: many of the utterances in it consist of only one sustained vowel. This is probably the worst scenario to show the advantages of dynamic modeling in front of the static one.

4. Conclusions and future work

As it probably could be expected, our HMM based system could not achieve the highly competitive baseline results. Nevertheless we believe that our results are close enough to them to be taken into consideration. Mainly because they are obtained using a completely different approach, not only to the baseline system but to most of the systems currently used for this kind of task. This leads us to two considerations: first, that perhaps further efforts in the optimization of our system may bring results closer to the baseline; and second, that such an approach could be particularly useful if embedded in systems based on global statistics, just as the baseline is.

With respect to further optimization of our system, we are not only considering finer adjust of several parameters that we could not study in full depth in time to adhere to this challenge, but also how to deal with those additional statistics and functionals that bring so good results to systems based on global statistics. We are also working on how to guarantee that the only source of difference in the likelihood scores is due to the paralinguistic cues involved, and not to differences in the phonetic contents or the path followed along the HMMs.
With respect to the embedment of HMM based systems in systems using global statistics and functionals, we consider this embedment is twofold due to the intrinsically different nature of the herein proposed approach: first, ensemble learning strategies, such as ROVER, can profit from such different kinds of knowledge; second, the frame likelihoods provided by the HMM engine can constitute the basis for designing new LLDs usable in frameworks based on Support Vector Machines or similar. In this last case, notice that these LLDs would be, by themselves, able to achieve results close to those achieved by the rest of them. As a matter of fact, what we are using in this paper can be seen as a linear classifier using as information the functional sum of the frame log-likelihoods.

5. Acknowledgements

This work has been funded by the Spanish Government’s Ministry of Science and Innovation through the SARAI project (TEC2010-21040-C02-01). The author also wishes to thank the INTERSPEECH 2011 Speaker State Challenge team for organizing the evaluation and providing the databases.

6. References


