Acoustic Group Feature Selection Using Wrapper Method for Automatic Eating Condition Recognition

Dara Pir\textsuperscript{1} and Theodore Brown\textsuperscript{2}

Department of Computer Science
\textsuperscript{1}The Graduate Center, \textsuperscript{2}Queens College
City University of New York, New York, USA
dpir@gradcenter.cuny.edu, tbrown@gc.cuny.edu

Abstract

In this paper, we present a wrapper-based acoustic group feature selection system for the INTERSPEECH 2015 Computational Paralinguistics Challenge (ComParE) 2015, Eating Condition (EC) Sub-challenge. The wrapper-based method has two components: the feature subset evaluation and the feature space search. The feature subset evaluation is performed using Support Vector Machine (SVM) classifiers. The wrapper method combined with complex algorithms such as SVM is computationally intensive. To address this, the feature space search uses Best Incremental Ranked Subset (BIRS), a fast and efficient algorithm. Moreover, we investigate considering the feature space in meaningful groups rather than individually. The acoustic feature space is partitioned into groups with each group representing a Low Level Descriptor (LLD). This partitioning reduces the time complexity of the search algorithm and makes the problem more tractable while attempting to gain insight into the relevant acoustic feature groups. Our wrapper-based system achieves improvement over the challenge baseline on the EC Sub-challenge test set using a variant of BIRS algorithm and LLD groups.

Index Terms: ComParE 2015, Eating Condition, wrapper method, acoustic group feature selection, computational paralinguistics

1. Introduction

Computational Paralinguistics (CP) deals with how something is said whereas Automatic Speech Recognition (ASR) is concerned with what is said [1, 2]. Starting in 2009, the INTERSPEECH paralinguistic family of challenges set out to provide a unified test-bed to allow for comparison of performances under the same exact conditions [3]. The ComParE 2015, EC Sub-Challenge uses the same unified acoustic feature set as the one used in the past two years across all the Sub-Challenges [4, 5, 6]. This feature set, generated by the openSMILE toolkit [7, 8], was the most effective [5] and the largest set (6373 features) used so far in these Challenges and provides a consistent test-bed across all the Challenges. The ComParE 2015, EC Sub-Challenge aims to classify which of the six types of food or no food is being eaten by the speaker using the audio tracks of the audio-visual iHEARu-EAT database [6]. Analyzing speech while eating, as a new field of study, has promising applications including health, security and ASR while eating [6].

High dimensional systems potentially include irrelevant features that degrade the system’s accuracy performance. Feature selection can help reduce this effect. Feature selection methods can be grouped into two main categories: the wrapper and filter methods [9]. The wrapper method uses accuracy scores generated by a classifier to evaluate feature subsets whereas the filter method uses general statistical characteristics of data for the evaluation. A wrapper-based method has two components: the feature subset evaluation and the feature space search. Our feature subset evaluation is performed using the SVM classifier. Since the feature space search is computationally intensive we use the fast BIRS algorithm. To further improve accuracy performance, a variant of the basic BIRS algorithm is presented. The EC Sub-Challenge uses Unweighted Average Recall (UAR) as performance measure.

The acoustic features are supra-segmental information obtained by applying, on the chunk level, statistical functionals like min/max of peak position to LLD contours like Mel-Frequency Cepstral Coefficients (MFCCs) [3, 10]. The feature space is considered in groups instead of individually. An LLD-based grouping is acoustically motivated. If an acoustic feature is relevant to a classification task then the collection of features generated by the application of functionals on that feature are also potentially useful for that classification task and vice versa. Accordingly, we partition the feature space into groups where each group is associated with a unique LLD.

The motivation for using group-based feature selection is explained by the following experiment. The top portion of Figure 1 depicts the classification score in UAR of the baseline accuracy scores generated by a classifier to evaluate feature subsets whereas the filter method uses general statistical characteristics of data for the evaluation. A wrapper-based method has two components: the feature subset evaluation and the feature space search. The feature subset evaluation is performed using the SVM classifier. Since the feature space search is computationally intensive we use the fast BIRS algorithm. To further improve accuracy performance, a variant of the basic BIRS algorithm is presented. The EC Sub-Challenge uses Unweighted Average Recall (UAR) as performance measure.

The acoustic features are supra-segmental information obtained by applying, on the chunk level, statistical functionals like min/max of peak position to LLD contours like Mel-Frequency Cepstral Coefficients (MFCCs) [3, 10]. The feature space is considered in groups instead of individually. An LLD-based grouping is acoustically motivated. If an acoustic feature is relevant to a classification task then the collection of features generated by the application of functionals on that feature are also potentially useful for that classification task and vice versa. Accordingly, we partition the feature space into groups where each group is associated with a unique LLD.

The motivation for using group-based feature selection is explained by the following experiment. The top portion of Figure 1 depicts the classification score in UAR of the baseline features individually using the EC Sub-Challenge baseline setup. Weka’s SMO classifier [11] with default settings. The bottom part of the figure displays the result of the same ex-
experiment using feature groups where each group contains fifty individual features. The similarities between the two graphs, together with the observation that the maximum peak heights of the feature groups are around 45% UAR vs. 25% for their individual feature counterparts, suggests that combining individual features within a group could improve system performance. This is not surprising. It is expected that the combination of single features could yield higher UAR scores. Moreover, the features are not arranged according to a random order. All of the features generated by the functionals within one LLD are visited before those of the next LLD. The LLDs, in turn, are arranged by family type. For example, all of the pem_f1Mag_n_fcc_sm family LLDs are visited before moving on to the audSpec_Rfilt_sma_family of LLDs.

Motivated by the above argument, we propose a wrapper-based feature selection system that processes meaningful feature groups and explore using LLD-based partitions as our groups.

The organization of the paper is as follows. Section 2 goes over the background and related work on feature space search algorithms and feature space partitioning. Section 3 gives the details on BIRS method in addition to our proposed variant of it. The experimental results are discussed in section 4 and the paper concludes with suggested future work in section 5.

2. Background

2.1. Feature Space Search

An exhaustive search of the feature space has exponential time complexity and is prohibitive when dealing with high dimensions. Heuristic methods are hence used to explore the feature space. We will consider some popular deterministic methods herein. For the purpose of facilitating comparison of time complexity among various algorithms we use wrapper evaluation cycles as the measure and unless otherwise mentioned will not discuss the time complexity of the learning algorithms. Wrapper-based feature selection methods are computationally intensive but usually give superior performance as they tend to select subsets that suit the biases of the learning algorithm. This demands the use of fast search algorithms to render the problem more tractable while still retaining the benefit of using the wrapper.

Best First Search (BFS) [9] is a popular and effective search that adds the best performing feature, from the subset of features not already selected, to the selected list. This process is continued either until there are no features left to consider or up to any user-specified point. BFS has quadratic growth and, depending on the stopping criteria, could be computationally impractical for large feature sets. [14] investigates Sequential Forward Selection (SFS) and Linear forward Selection (LFS) searches which also have quadratic growth. SFS is a hill-climbing search similar to BFS but the search is terminated when the evaluation score is not improved by any single feature addition. LFS makes SFS computationally tractable by limiting the number of features considered by means of a user-specified parameter. The drawback of this method is that features with low scores that may improve the system's accuracy performance are eliminated from the search.

In this paper, we will explore two forward search type linear algorithms: Rank Search (RS) [14] and Best Incremental Ranked Subset (BIRS) [13]. Each of these two algorithms is performed in two steps. In the first or ranking step, the features are evaluated using either a filter or a wrapper and then ranked by the evaluation score from highest to lowest. For the second or feature subset selection step, each algorithm uses a different method to find the optimal feature subset. We note that the first step can be performed much faster than the second one.

In the second step of RS, we start by including the best, i.e., highest ranking, feature to the selection subset. Then we add the next best feature to the subset and evaluate the resulting subset using our wrapper. We continue this process with the third best feature all the way until there are no more features left while evaluating the subsets at each addition. The best subset is one yielding the highest evaluation score. This is a fast algorithm with $2 \times N$ evaluations and, in general, creates large subsets.

In the second step of BIRS, a similar algorithm to RS, we construct the optimum feature subset list as follows. We start by adding the best feature to the subset list. Next we add the next best feature to the subset list and accept the addition only if the resulting subset is evaluated to a higher UAR, by a threshold level, than the current one. We continue this selection process until the ranked list is traversed once completely. The algorithm performs $2 \times N$ evaluations and, in general, generates smaller subsets as compared to RS. Similar to the quadratic growth searches mentioned earlier, both RS and BIRS can be stopped before completion. We will not, however, consider this option in our experiments.

Linear methods such as RS and BIRS are still computationally intensive if the space search uses individual features. Alternatively, using a group-based search reduces a linear time algorithm's $N$ evaluation cycles, in the second step, to $k$ cycles where $k$ is the number of groups. For groups partitioned by LLDs, the number of cycles is 130 as opposed to 6373 for the case with no partitioning. We note that the time complexity of the learning algorithm itself increases with the size of the subset and therefore each evaluation cycle is not performed with the same speed.

2.2. Use of Domain Knowledge

Domain knowledge has been used to make meaningful partitions using LLDs in [15]. The LLD partitions are used in a multi-view [16] approach feature reduction system. The authors use the filter approach and show the superiority of their system to the single-view counterpart. Groups also have been used in another context involving studies on genes. For example, when predefined groups contain predictors it may be beneficial to select members of a whole group together [17].

3. Method

3.1. Feature Subset Evaluation

We use the WEKA data mining toolkit’s SMO and LIBLINEAR [18] SVM classifiers to implement our wrapper-based group feature selection system. All the experiments on development data use Leave-One-Speaker-Out Cross-Validation (LOSO-CV). The results on test data are obtained using the setup used for the calculation of the Sub-Challenge baseline with the SVM complexity (C) parameter as variable.

3.2. Feature Space Search

Two different systems are used for the feature space search: the basic BIRS and a variant of it. The advantage of BIRS-based system is its speed and its flexibility to model various behaviors through setting of a parameter.
Algorithm 1 Modified group-based basic BIRS for wrapper style feature selection with LOSO-CV

Input: Group: feature group list, C: classifier, T: acceptance threshold level
Output: BestScore, BestSubset

1. RankedGroup ← {}
2. for each Group, ∈ Group do
3. Score ← WrapperClassify (Group, C)
4. append Group, to RankedGroup according to Score
5. end for
6. BestScore ← 0
7. BestSubset ← {}
8. for i ← 1, k do ▷ k: RankedGroup list size
9. TempSubset ← BestSubset ∪ RankedGroup,
10. Score ← WrapperClassify (TempSubset, C)
11. if Score > BestScore then
12. BestScore ← Score
13. BestSubset ← TempSubset
14. end if
15. end for

3.2.1. Basic BIRS

The basic BIRS algorithm was described in the previous section. Algorithm 1 depicts the detailed steps of the algorithm for the group-based version. The same wrapper is used for both ranking and subset selection steps. The ranking may be accomplished by using a filter or a different wrapper than one used for subset selection.

The algorithm receives as input a list of feature groups, a classifier and an acceptance threshold parameter denoted by T. The current classifier score replaces the best score if it is greater than the best score by the threshold amount. We note that the WrapperClassify() function in Algorithm 1 performs the LOSO-CV. No statistical measurements are performed for threshold estimation and we simply use a fixed value for it.

Broadly, there are three ranges of values for the T parameter. A very small value of $\epsilon = 10^{-6}$ would include a group if its inclusion improves the system UAR. This tends to make a significant dimensional reduction in general. A system with a larger value for T would be even more selective in accepting groups into the selected list. This results in greater dimensional reduction. It is also possible to use a negative value for T. This is the approach we take in our experiments. Whereas a large positive T tends to select the very best performing feature groups, a large negative T will reject only the worst performing, i.e., the highly irrelevant, ones.

The choice of using negative values for T is motivated by the following argument. The types of sounds generated by speaking while eating various food categories span a great range and many features are required for their analysis. Furthermore, we are using group-based feature selection and removing whole groups could easily degrade our system performance. Our aim, therefore, is to improve UAR performance by attempting to remove the most highly irrelevant feature groups. Dimensional reduction per se is not our primary objective.

3.2.2. BIRS Variant

In addition to the basic BIRS approach, we explore using a variant of the basic algorithm. The detailed steps of the BIRS variant method are shown in Algorithm 2. In BIRS variant, instead of accepting the entire selected list returned by basic algorithm we select up to and including the feature group having the best UAR performance. The BIRS variant presented here combines elements of two linear search algorithm, RS and BIRS, into a single algorithm. As in the case of basic BIRS, the parameter T setting allows for modeling various behaviors. To increase the generalization power of the algorithm, instead of simply selecting the feature group with the highest UAR, we select the feature group that results in the highest value for the following expression:

$$UAR_{Score} = \alpha \times STD$$

where $STD$ is the standard deviation of the UAR list generated by LOSO-CV and $\alpha$ can be experimentally determined. We use a value of $\alpha = 0.2$ for our system.

Algorithm 2 Group-based BIRS variant for wrapper style feature selection with LOSO-CV

Input: Group: feature group list, C: classifier, T: acceptance threshold level
Output: BestAdjustedScore, BestAdjustedSubset

1. RankedGroup ← {}
2. for each Group, ∈ Group do
3. Score ← WrapperClassify (Group, C)
4. append Group, to RankedGroup according to Score
5. end for
6. BestScore ← 0
7. BestSubset ← {}
8. for i ← 1, k do ▷ k: RankedGroup list size
9. TempSubset ← BestSubset ∪ RankedGroup,
10. Score ← WrapperClassify (TempSubset, C)
11. if Score > BestScore then
12. BestScore ← Score
13. BestSubset ← TempSubset
14. end if
15. end if
16. if AdjustedScore > BestAdjustedScore then
17. BestAdjustedScore ← AdjustedScore
18. BestGroup ← Group,
19. end if
20. end for
21. generate BestAdjustedSubset from BestSubset and BestGroup.

Figure 2: Results of feature group selection on development data. The system is evaluated using SVM C parameters ranging from 0.0005 to 0.02 in approximately double increments. The BIRS T parameter is set to -1.
Table 1: Results of the basic BIRS algorithm on test data. Complexity parameter of SVM, T: Acceptance threshold parameter. GR: Number of groups removed in the feature selection. FR: Number of the individual features removed in the feature selection. UAR: UAR performance of the system in %.

<table>
<thead>
<tr>
<th>C</th>
<th>T</th>
<th>GR</th>
<th>FR</th>
<th>UAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>-1</td>
<td>4</td>
<td>200</td>
<td>66.5</td>
</tr>
<tr>
<td>0.0005</td>
<td>-1.5</td>
<td>8</td>
<td>347</td>
<td>67.6</td>
</tr>
</tbody>
</table>

Table 2: Results of the BIRS variant algorithm on test data. Columns are as described for Table 2.

<table>
<thead>
<tr>
<th>C</th>
<th>T</th>
<th>GR</th>
<th>FR</th>
<th>UAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0005</td>
<td>-1</td>
<td>8</td>
<td>347</td>
<td>67.6</td>
</tr>
<tr>
<td>0.0005</td>
<td>-1.5</td>
<td>1</td>
<td>39</td>
<td>67.9</td>
</tr>
</tbody>
</table>

4. Experimental Results

4.1. Basic BIRS Results on Development Data

We start our wrapper-based group feature selection experiment by exploring the basic BIRS search algorithm. The system is evaluated on development data using SVM complexity (C) parameters ranging from 0.0005 to 0.02 in approximately double increments, i.e., 0.0005, 0.001, 0.002,..., 0.02. The BIRS T parameter is set to -1. LIBLINEAR is used with default parameters on standardized data as in the Sub-Challenge baseline calculation. The results are displayed in Figure 2.

4.2. Basic BIRS Results on Test Data

We select the system with best UAR performance on development data, i.e., one using C = 0.001, for the classification of test data. Using the feature subset determined in the development stage we train and classify following the same setup as in the Sub-Challenge baseline experiment. We reach a UAR value of 66.5% which is 0.6% above the baseline result of 65.9%. The feature selection has removed four LLD groups that contain 200 individual features. All of the removed groups have UAR rankings within the top 70 of 130 total.

Extending the range of the SVM C parameters we repeat the process with C = 0.0003. We reach a UAR of 67.2% on test data, a 1.3% improvement over the baseline result. The feature selection has removed two LLD groups that contain 347 individual features. All of the removed groups have UAR rankings within the top 70 of 130 total.

On test data of 67.9%, an improvement of 2% over the baseline result. The feature selection has removed only one LLD group that contains 39 individual features. The LLD group removed is jitterLocal_sma which has the lowest UAR ranking of all LLD groups.

4.4. Feature Group Ranking

Tables 3 and 4 display the list of five groups having, respectively, the highest and the lowest UAR rankings in our best performing experiment that uses BIRS variant with parameters $C = 0.0005$ and $T = -1.5$. We note that groups with smaller number of features tend to have lower UAR rankings. As when dealing with individual features, lower rank does not necessarily indicate less relevance. Our greedy search algorithms, however, visit groups in descending UAR ranking order.

5. Conclusions and Future Work

In this paper, we present wrapper-based feature group selection methods that combine two techniques to render a computationally intensive problem more tractable. Firstly, we make use of the fast and flexible linear search algorithm Best Incremental Ranked Subset as well as a variant of it that uses the final subset selection criteria of Rank Search. Secondly, we partition the feature space into meaningful groups based on Low Level Descriptors. Group processing reduces the number of wrapper evaluation cycles of our linear algorithms significantly.

Our best performance is obtained using the novel wrapper-based approach that combines Low Level Descriptor groups with BIRS variant search for Computational Paralinguistic data classification. We reach 67.9% UAR on Eating Condition Sub-Challenge, improving the baseline performance by 2%.

A natural extension of our technique is to incorporate intra-group feature selection into our inter-group approach. Future work could also include comparing accuracy results of our feature group selection methods with those obtained by the computationally intensive individual feature selection counterparts.
6. References


