# Using Machine Learning to Cope with Imbalanced Classes in Natural Speech: Evidence from Sentence Boundary and Disfluency Detection

*Yang Liu*[1,3] *Elizabeth Shriberg*[1,2] *Andreas Stolcke*[1,2] *Mary Harper*[3]

[1]International Computer Science Institute, USA [2]SRI International, USA
[3]Purdue University, USA

{yangl,ees,stolcke}@icsi.berkeley.edu, harper@ecn.purdue.edu

## Abstract

We investigate machine learning techniques for coping with highly skewed class distributions in two spontaneous speech processing tasks. Both tasks, sentence boundary and disfluency detection, provide important structural information for downstream language processing modules. We examine the effect of data set size, task, sampling method (no sampling, downsampling, oversampling, and ensemble sampling), and learning method (bagging, ensemble bagging, and boosting) for a decision tree prosody model. Results show that (1) bagging benefits both tasks, but to different degrees, (2) the benefit from ensemble bagging decreases as data size increases, and (3) boosting can outperform bagging under certain conditions.

## 1. Introduction

Current speech recognition systems leave out structural information about the location of punctuation and disfluencies, which is often assumed to be available for downstream language processing modules. This paper investigates the use of machine learning methods for the detection of sentence boundaries (SU)[1] and disfluency interruption points (IP) given the speech signal and its transcription. The following example shows a transcription with SU boundary and IP events annotated.

```
yeah <SU> we really haven't tried
camping with <IP> with our daughter
yet <SU> we <IP> we'd like to now that
she's getting a little bit older <SU>
```

In the training data, SU boundaries and IPs are handmarked by annotators using both word transcripts and the recorded audio [1]. In testing, given a word sequence (provided by human transcription or speech recognition output) $W_1$ $W_2$ ... $W_n$ and the speech signal, the task is to determine a class for each inter-word boundary. The possible orthogonal classes we consider are (1) SU boundary, (2) IP, or (3) neither (fluent, within-SU).

For transcriptions, we can use either human transcriptions or the output of a speech recognizer. Because we focus here on gains from machine learning for a prosodic classifier, before and after combination with a language model, we have chosen to report results using hand-transcribed words throughout. This represents a "best-case" scenario for the language model, and factors out the impact of recognition errors on our investigation.

Our general approach [2] has three components: a prosody model, a hidden event language model, and a method for combining the models. The prosody model is a word-boundary classifier. Each word has an associated feature set, and the classifier learns to discriminate SU boundaries from non-SU boundaries, or IPs from non-IPs by using these features. The features reflect prosodic information, including pause, duration, pitch, and energy patterns. In this paper we focus on building better classifiers using the current feature set rather than finding better prosodic features. We use decision trees for our prosody model because they perform well, can combine categorical and continuous features (including those that are partially undefined), and are interpretable. A hidden event LM [3] models the joint distribution of boundary types and words in a hidden Markov model (HMM) with the hidden variable being the boundary type. An integrated HMM approach models the joint distribution $P(W, F, E)$ of the word sequence $W$, prosodic features $F$, and hidden event types $E$.

For both the SU and IP detection tasks, we encounter an imbalanced data set problem, because these events are much less frequent than nonevents. This causes classifiers to "ignore" inherent properties of the smaller class. Thus, we need some way to help our classifiers cope with the highly skewed distributions. Although the machine learning community investigated methods for addressing the imbalanced data set problem, e.g., [4], our data is different from many data sets studied in the machine learning community in that it uses speech, the data set is large and noisy, and the results from the classifier (i.e., the prosody model) are combined with an LM for final decisions.

This paper is organized as follows. Section 2 describes the machine learning techniques and our experimental set up for the two tasks. Section 3 reports results for each task. Section 4 summarizes our findings. A companion paper [5] describes our full SU and IP detection systems, and reports results for recognized words and other metadata tasks.

[1]In spontaneous speech "sentences" are different from written text. We use "SU" to represent these sentence-like units. See [1] for details.

## 2. Methods and Experimental Setup

### 2.1. Approaches Used for SU and IP Detection

Previous work on a small data set [6] investigated a variety of sampling approaches, as well as bagging and ensemble methods for detecting SU boundaries. In the present study we examine the impact of using more data, and compare results on the SU detection task to results using the same machine learning techniques on the new task of IP detection. The following is a brief description of the machine learning approaches we apply to both tasks:

- Sampling:
  - *original set*: use the original training set as it is, i.e., no sampling is used.
  - *downsampling*: randomly sample the majority classes to have the same number of instances as the minority class.
  - *oversampling*: replicate the minority classes to match the number of instances in the majority class.
  - *ensemble sampling*: split the majority class into $N$ sets, each of which is combined with all of the minority class samples to make a balanced training set to train a classifier. The final decision is made by combining all $N$ classifiers.
- Bagging:
  - *bagging on downsampled set*: Bagging [7] combines different classifiers that are trained from different samples from a training set (with replacement). We applied bagging to the downsampled training set.
  - *bagging on ensemble sampling*: for each balanced training set formed from the ensemble sampling approach, apply bagging. The final result interpolates the output from all the classifiers.
- Alternating Decision Tree Boosting: On many machine learning tasks, the application of boosting to decision trees has resulted in improved classification accuracy. Boosting [8] combines multiple weak learning algorithms. Each classifier is built based on the output of the previous classifiers, mostly by focusing on the samples for which the previous classifiers made incorrect decisions. In contrast to bagging, boosting generates classifiers sequentially, and thus it cannot be implemented in parallel and is computationally more expensive. Freund and Mason [9] proposed an alternating decision tree (ADT) learning algorithm based on boosting that produces a single tree that is a generalization of the decision trees. We applied this ADT approach to SU boundary and IP detection tasks.

| | SU | | IP |
|---|---|---|---|
| | Small set | Large set | Large set |
| Training set | 128K | 428K | 428K |
| Test set | 16K | 53K | 53K |
| Percentage of minority event (%) | 13.0 | 13.56 | 4.54 |

Table 1: Statistics on the data sets used for the SU and IP detection tasks. The small set used in the previous study [6], shown in the second column, is a subset of the large set used in this paper.

### 2.2. Experimental Setup

We used data from the Switchboard conversational telephone speech corpus [10]. The corpus in our experiments is the training data used for the 2003 Fall DARPA Rich Transcription evaluation. We split the 754 conversations into training and testing sets. These conversations were annotated with SUs and IPs [1]. The same conversations are used for both the SU and IP tasks. Table 1 shows the experimental setup, including the training and testing set sizes (number of inter-word boundaries) and the percentage of the minority class in the data set for each task. For comparison, we also include the data description for the smaller set used in the previous investigations of the SU task [6].

We started with about 100 prosodic features representing duration, pitch, and energy. For each task, we trained a decision tree from a randomly downsampled training set, and then used only the features selected by this decision tree for the other sampling or bagging approaches. This was done in order to minimize computational effort. Note that this feature selection approach is suboptimal; other more sophisticated techniques may be able to select a better feature subset.

We evaluate performance using classification error rate, which is defined as the ratio of word boundaries that are classified incorrectly to the total number of word boundaries. When using speech recognition output for the experiments, due to insertion, deletion, and substitution word errors, it is not straightforward to align the reference SU boundaries or IPs with the system output. This is one of the reasons why we choose to report results on the reference transcriptions and factor out alignment errors. In the official NIST-EARS evaluation of the SU and IP tasks a different but correlated evaluation metric is used. See http://www.nist.gov/speech/tests/rt/rt2003/fall/ for details of that metric.

## 3. Experimental Results and Discussion

### 3.1. SU Detection Sampling and Bagging Results

Table 2 shows SU detection results using both the prosody model alone and in combination with the LM. If training uses a sampled set that differs from the test set in class distribution, the posterior probabilities from the decision tree need to be adjusted accordingly [6]. We include the results from our previous study [6] on the smaller corpus in order to examine the impact

| Method | | Small set (LM alone 5.02%) | | Large set (LM alone 5.27%) | |
|---|---|---|---|---|---|
| | | Prosody alone | Prosody+LM | Prosody alone | Prosody+LM |
| sampling | original | 7.33 | 4.08 | 7.45 | 4.53 |
| | downsampled | 8.48 | 4.14 | 8.05 | 4.42 |
| | oversampled | 10.67 | 4.39 | 8.46 | 4.64 |
| | ensemble sampling | 7.61 | 4.18 | 7.86 | 4.47 |
| bagging | on downsampled | 7.10 | 3.98 | 7.26 | 4.29 |
| | on ensemble | 6.93 | 3.89 | 7.22 | 4.35 |

Table 2: SU detection results in error rate (%). LM is trained from the original training set without any sampling.

of the data size on the sampling and bagging results.

As the data set size increases, we expected that the gain from using the original training set might be lost and the benefit from ensemble sampling might decrease, since the downsampled training set might be more representative of the data set. Table 2 shows that contrary to our expectation, using the original training set yields the best results, although it has a greater cost in training time. As expected, the gain from ensemble sampling is diminished as the data set size increases. When the data set is small, ensemble sampling has the advantage of making use of the full data set within the ensemble. As the data set increases and is inherently more representative, the ensemble benefit decreases.

Similar to using the smaller data set, oversampling is computationally expensive and does not yield a performance improvement. Downsampling the training set performs reasonably well, and has the advantage of saving computation. This is important when the training set size is large, i.e., hundreds of thousands of data samples.

For both data sets, bagging outperforms the single classifier. This shows that the combination of multiple classifiers for this task can reduce the variance relative to that of a single classifier or that in a randomly sampled training set.

Notice from the table that the combination of prosody model and LM achieves better performance than using either knowledge source alone. However, the gain from applying a sampling or bagging method on the prosody model may be diminished after combination with the LM.

### 3.2. IP Detection Sampling and Bagging Results

Table 3 shows results of the sampling and bagging approaches for the IP detection task. In addition to the results on the original test set, we show results on the downsampled test set when using the prosody model alone.

If the original training set is used, then because the IP samples comprise an extremely small portion of the training set, the decision tree does not split. The classifier is not able to learn the characteristics of the minority class. Therefore the classifier performs at chance on the original test set and does not provide any information when it is combined with the LM. This differs from the SU detection task, in which the best performance is achieved by using the original training set among the different sampling approaches. However, when the downsampled training set is used, the IP classifier performs substantially better than chance on the downsampled test set.

| Method | | Prosody alone | | Prosody+LM |
|---|---|---|---|---|
| | | DS | Original | Original |
| sampling | original | 50 | 4.36 | 2.34 |
| | downsampled | 23.76 | 4.36 | 2.27 |
| | oversampled | 27.69 | 4.36 | 2.31 |
| | ensemble | 22.07 | 4.36 | 2.24 |
| bagging | on DS | 20.64 | 4.36 | 2.25 |
| | on ensemble | 20.20 | 4.36 | 2.24 |

Table 3: IP detection results in error rate (%). Chance performance is 4.36% on the original test set. The error rate of using LM alone is 2.34%. 'DS' denotes 'downsampled'.

As with results for the SU detection task, bagging and ensemble bagging perform significantly better than the other approaches on the downsampled test set when using the prosody model alone. Yet on the original test set, when the priors are taken into consideration, none of the approaches (downsampling, bagging, ensemble) is able to beat the bias of the majority class. Despite achieving only chance performance on the IP detection task when used alone, the prosody model provides added information after it is combined with the LM. However, the relative error rate reduction is smaller for the IP detection task than for the SU detection task, i.e., 4.3% versus 18.6% respectively.

Figure 1 shows the ROC curves for the IP and SU detection tasks for the original test set using the downsampled training set, bagging, and ensemble bagging. These curves suggest that bagging indeed improves the performance over using a single randomly downsampled training set. The ROC curve from ensemble bagging is similar to that using bagging on one downsampled set. Notice also from the curves that the improvement on the IP detection task is larger than on the SU task, suggesting that bagging improves the generality of decision tree classifiers more on the noisy IP task than on the SU task.

### 3.3. ADT Boosting for SU and IP Detection Tasks

Results using the ADT algorithm for the prosody model alone are shown in Table 4. The model is trained using the downsampled training set and tested on the downsampled test set. Since the algorithm does not generate posterior probabilities of

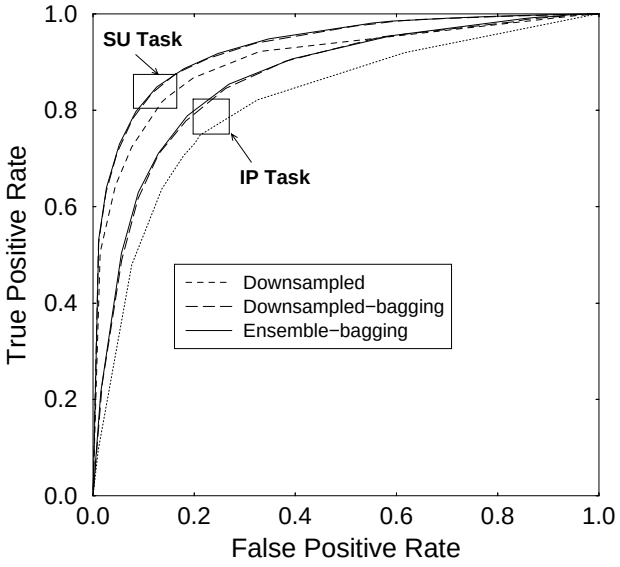


Figure 1: ROC curves for IP and SU detection using the prosody model alone.

| | Bagging | Boosting ADT |
|---|---|---|
| SU | 14.3 | 14.8 |
| IP | 20.6 | 19.3 |

Table 4: SU and IP detection results in classification error rate (%) using the ADT learning algorithm and bagging.

class membership for each test sample,[2] we do not report results on the true test using the prosody model alone, or when combining the prosody model with the LM, since both conditions require classifiers that output posterior probability estimates. Compared to the performance of bagging, the ADT boosting algorithm improves performance on the IP task but not on the SU detection task. This highlights the difference between the SU and IP tasks, suggesting that the metric of reducing classification errors used by the ADT learning algorithm may be better for the noisy IP task, while the information gain used in classical decision tree learning is more appropriate for the SU task.

## 4. Conclusions

We have examined machine learning techniques for addressing the imbalanced data problem in two spontaneous speech tasks. We investigated the impact of data set size on sampling approaches for an SU detection task. This is extremely important since most speech processing problems have a large training set and finding the best sampling and bagging approach is crucial to building a good classifier. Similar results were found when the data size increases; however, the gain from ensemble sampling diminishes as the data set is enlarged. Our investigation of the IP task highlights differences between the IP and SU tasks, which could be due to differences in the magnitude of skew, inherent differences in cues to the phenomena, or both. We found that sampling techniques are more important in the case of the IP task, where the data skew problem is much more severe. Bagging approaches substantially improve the accuracy of both SU and IP detection. Finally, our initial investigation of boosting with alternating trees highlights additional differences between the SU and IP detection tasks.

[2] In future work, we will investigate methods for converting the score of the ADT learning algorithm to a posterior probability.

## 5. Acknowledgments

The authors gratefully thank Yoav Freund for his guidance in applying the ADT learning algorithm to this task. This research has been supported by DARPA under contract MDA972-02-C-0038, NSF-STIMULATE under IRI-9619921, and NASA under NCC 2-1256. Distribution is unlimited. Any opinions expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA, NSF, or NASA. Part of this work was carried out while the last author was on leave from Purdue University and at NSF.

## 6. References

[1] S. Strassel, “Simple Metadata Annotation Specification V5.0”, Linguistic Data Consortium, 2003.

[2] E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur, “Prosody-based Automatic Segmentation of Speech into Sentences and Topics”, Speech Communication, pp. 127-154, 2000.

[3] A. Stolcke and E. Shriberg, “Automatic Linguistic Segmentation of Conversational Speech”, Proc. ICSLP, pp. 1005-1008, 1996.

[4] ICML, Workshop on Learning from Imbalanced Datasets II, 20th International Conference on Machine Learning, 2003.

[5] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, B. Peskin, and M. Harper, “The ICSI-SRI-UW Metadata Extraction System”, To Appear in Proc. ICSLP 2004.

[6] Y. Liu, N. Chawla, E. Shriberg, A. Stolcke, and M. Harper, “Resampling Techniques for Sentence Boundary Detection: A Case Study in Machine Learning from Imbalanced Data for Spoken Language Processing”, Technical Report, ICSI, 2003.

[7] L. Breiman, “Bagging Predictors”, Machine Learning, 24(2), pp. 123-140, 1996.

[8] Y. Freund, “Boosting a Weak Learning Algorithm by Majority”, Information and Computation, pp. 256-285, 1995.

[9] Y. Freund and L. Mason, “The Alternating Decision Tree Learning Algorithm”, Proc. ICML, pp. 124-133, 1999.

[10] J. Godfrey, E. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone Speech Corpus for Research and Development”, Proc. ICASSP, pp. 517-520, 1992.