PITCH ACCENT PREDICTION USING ENSEMBLE MACHINE LEARNING

Xuejing Sun

Department of Communication Sciences and Disorders, Northwestern University
2299 N. Campus Dr., Evanston, IL 60208, USA
sunxj@northwestern.edu

ABSTRACT

In this study, we applied ensemble machine learning to predict pitch accents. With decision tree as the baseline algorithm, two popular ensemble learning methods, bagging and boosting, were evaluated across different experiment conditions: using acoustic features only, using text-based features only; using both acoustic and text-based features. F0 related acoustic features are derived from underlying pitch targets. Models of four ToBI pitch accent types (High, Down-stepped high, Low, and Unaccented) are built at the syllable level. Results showed that in all experiments improved performance was achieved by ensemble learning. The best result was obtained in the third task, in which the overall correct rate increases from 84.26% to 87.17%.

1. INTRODUCTION

Prosodic events embody rich linguistic information that is critical for speech communication process. Many systems have been proposed to describe various prosodic patterns using a finite set of symbols (e.g. ToBI [9]). Automatic prediction of these symbols with high accuracy could therefore be useful in text-to-speech, automatic speech recognition, and corpus development. Depending on the application, prosodic event recognition systems can utilize acoustic information, text information, or from both.

A variety of algorithms have been investigated for predicting prosodic patterns, including Hidden Markov Model (HMM) (e.g.[2]), neural network (e.g. [6]), dynamical system [7], and decision trees (e.g. [5]). In the present paper, we explore the use of ensemble machine learning technique to predict ToBI [9] style pitch accents. For classification problems, ensemble learning algorithms construct a set of classifiers and then classify new data by taking a (weighted) vote of their predictions [3]. Lately, this approach has received much attention, and has been shown to be superior to single-classifier systems in many real world problems. Among various ensemble learning methods, bagging [1] and boosting [4] are probably the two most popular ones due to their effectiveness and ease of implementation.

In general, ensemble learning methods are algorithm-independent, and impose no restrictions on the choice of the basic learner. In this work, we chose *classification and regression trees* (CART) as our basic learning algorithm because it features: (1) faster training and testing compared with other algorithms (e.g., neural networks); (2) less hand-tuning of the parameters; (3) human-readable results; (4) easy application of the trained models to existing systems.

The paper is organized as follows. First we describe ensemble learning methods, specifically bagging and boosting. Then we present several experiments on pitch accent prediction with CART, bagging, and boosting. Finally, we discuss the results and present concluding remarks.

2. ENSEMBLE MACHINE LEARNING

2.1. Bagging

Bagging (Bootstrap Aggregation) [1] generates multiple classifiers by manipulating the training set. Each time a different training set is presented to the learning machine. The new training set is constructed by drawing samples from the original training set randomly with replacement. The final results are obtained usually by voting for classification or taking average for regression. For bagging to be successful, the learning machine should be *unstable*, that is, a small change in the training set would result in large changes in the training output. Decision tree and neural network are typical unstable learners.

2.2. Boosting

Boosting, specifically AdaBoost [4], also combines multiple classifiers by presenting different training set to the base learner. However, instead of using random selection as in bagging, the construction of a new training set depends on a weight distribution, which is updated over iterations. Initially all the training samples have the same weight. After each iteration, the weight distribution is updated such that misclassified samples have more weight. With the updated weight distribution, there are two ways of generating new training samples. In *reweighting*, the original training set is used, but each sample is associated with a new weight. This method is applicable to the learners that can handle weighted samples. In *resampling*, the new training set is constructed according to the weight distribution, where samples with more weights are more likely to be selected. Although it might be suboptimal, we used resampling in this work as an initial attempt, since it is easier to implement. Finally, in predicting a new sample, a weighted combination of multiple classifiers are used. Figure 1 illustrates the AdaBoost.M1 algorithm described by Freund and Schapire [4], an extension of the original boosting algorithm for multi-class problems.

Input: sequence of N training examples $((x_1, y_1), \ldots, (x_N, y_N))$
Initialize weight distribution $W_i=1/N$, where $i=1,\ldots, N$.
Do for t=1, … , T where T specifies the total number of iterations

1. Train classifier using weight distribution W_i
2. Get back a hypothesis $h_t : X \rightarrow Y$
3. Calculate the error of h_t:

$$\varepsilon_t = \sum_{i=1}^{N} p_i^t \left|\mathrm{sgn}[h_t(x_i) - y_i]\right|$$

$$\text{where } \mathrm{sgn}(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$$

if $\varepsilon_t > 0.5$, then set $T = t - 1$ and abort loop.

4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$
5. Set the new weights vector to be

$$w_i^{t+1} = w_i^t \beta_t^{1-|\mathrm{sgn}[h_t(x_i)-y_i]|}$$

Output the hypothesis

$$h_f(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) \left|\mathrm{sgn}[h_t(x) - y]\right|$$

Figure 1: Boosting algorithm AdaBoost.M1

2.3. Bias and variance

Why does ensemble learning work? It has been shown that the prediction error of a classifier can be decomposed into two components: bias and variance [1]. Ensemble methods like bagging can reduce the amount of variance. Boosting can reduce both bias and variance. Individual decision trees have high variance in terms of generalization accuracy. Thus, applying ensemble learning on decision trees can improve performance by lowering variance.

3. EXPERIMENTS

3.1. The corpus

Training and testing data were taken from Boston University Radio Speech Corpus, speaker F2B. The database, consisting of about 40 minutes speech read by a female professional announcer, is labeled using the ToBI [9] system. Similar to Ross and Ostendorf [7], the ToBI pitch accent labels were grouped into four types: High, Low, Down-stepped high, and Unaccented. The labels were aligned with syllables. The distribution of pitch accent types in the database is shown in Table 1. The database also provides text information, such as part-of-speech, and acoustic information such as segment duration. F0 values were determined by the SHRP algorithm[11]. The data set was split into training and testing sets with approximately a 4:1 ratio.

	Pitch accent type			
	Unaccented	**High**	**Downstep**	**Low**
Training set	7804	2717	853	151
Testing set	1929	677	211	35

Table 1: Pitch accent distribution in the database

3.2. Building models

In this work, we conducted three experiments to evaluate ensemble learning: (1) pitch accent prediction using only acoustic features; (2) pitch accent prediction using only text features; (3) pitch accent prediction using both acoustic and text features. Note that, similar to [7][8], we predicted pitch accent at syllable level, which assumes the syllable boundaries are known. In each experiment, we built models using single CART, bagging with CART, and AdaBoost with CART. The number of iterations for bagging and boosting was limited to 50. Guided by the theory of bias and variance decomposition, we applied ensemble learning as follows: Overtrain CART to generate a tree with low bias by using a small stop value, which refers to the minimum number of samples in the leaf nodes; Use bagging or boosting to reduce variance. "WAGON" [12] program, an implementation of standard CART, was used to build classification trees

3.2.1. Pitch accent prediction using acoustic features

Many acoustic features are thought to be correlates of pitch accent. Only fundamental frequency (F0), energy, and segmental duration were considered in this study. The F0 related features were derived from the so-called underlying pitch targets [13]. Below we describe the pitch target analysis procedure briefly, and the details can be found in [10].

First, for each syllable we define

$$T(t) = at + b \quad (1)$$

$$y(t) = \beta \exp(-\lambda t) + at + b \quad (2)$$

where $T(\cdot)$ represents the underlying pitch target, and $y(\cdot)$ represents the surface F0 contour. Coefficient β is a scaling parameter, and its value is the distance between F0 contour and the underlying pitch target when $t = 0$. Parameter λ is a positive number representing the rate of decay of the exponential part. Parameters a and b are the slope and intercept of the underlying pitch target.

Next, let (t_0, y_0) denote the first point on the F0 contour, and let (t_1, y_1) denote a point where underlying pitch target has been approached, then we have:

$$y(t) = (y_0 - y_1 + at_1)\exp(-\lambda t) + at + y_1 - at_1 \quad (3)$$

The parameters of the model are estimated by nonlinear regression. When nonlinear regression fails, linear regression is performed. In practice, for (t_0, y_0), we use an average of the first two F0 values in estimation because the first point can be aberrant. For (t_1, y_1), we use the point in the middle of a segment, which seems to work best.

In constructing the feature set, we extracted two parameters from each pitch target, middle F0 value (MidF0) and the slope. We also computed the change of F0 and slope between pitch targets, i.e. Δ MidF0 and Δ Slope. Together with syllable energy and duration, the feature set contains:

- MidF0 of the current, previous, and next pitch target
- Δ MidF0 with respect to the previous and next pitch target
- Slope of the current, previous, and next pitch target
- Δ Slope with respect to the previous and next pitch target
- Syllable duration
- Syllable energy

Stop value 30 was chosen for single CART since it yielded low error on the testing set. For bagging and boosting, stop value 5 was used in order to generate overtrained trees with low bias.

3.2.2. Pitch accent prediction using text features

Predicting pitch accent from text has been studied extensive in the past due to its critical role in text-to-speech systems. It has been shown that many factors can affect pitch accent placement. In this work, however, we limited our choices to those that could be derived from unrestricted text without much difficulty. The feature set contains:

- Vowel identity
- Syllable stress of the previous and next syllable
- The position of the current, previous, and next syllable in a word
- Number of syllables in the current and previous word
- Part-of-speech of the previous and next words
- A composite feature made up by part-of-speech and stress for the current syllable
- Number of words from the beginning of the sentence and to the end of the sentence

The stop value was 20 for single CART, and 5 for bagging. For boosting, however, stop value 20 was used, which gave better results than a smaller value.

3.2.3. Prediction with both acoustic and text information

In this experiment, we combined the acoustic and text features listed in the last two sections to predict pitch accent. The stop value was 20 for single CART, and 5 for both bagging and boosting.

3.3. Results

To facilitate a quick comparison, Table 2 lists the overall correct rate regardless of pitch accent type for all the experiment conditions. Detailed evaluation results in the form of confusion matrix are shown in Tables 3-11. In the tables, each column represents the prediction results for each pitch accent type with percentage and frequency count. We adopted the same evaluation method used by Ross and Ostendorf [7][8] since those studies and the present work are very similar with respect to the experiment configuration.

	Overall correct rate (%)
Acoustic-CART	82.89
Acoustic - Bagging with CART	84.71
Acoustic - AdaBoost with CART	84.71
Text - CART	80.47
Text - Bagging with CART	80.64
Text - AdaBoost with CART	80.50
Both - CART	84.26
Both - Bagging CART	86.89
Both - AdaBoost with CART	87.17

Table 2: The overall correct rate of CART, bagging, and AdaBoost

It can be seen from Table 1 that ensemble learning can indeed yield favorable results than a single decision tree. The improvement is most significant in the third task, in which both acoustic and text features were used. This implies that when more input features are available, their usefulness might be better exploited by combining multiple machines. In the second task, the improvement seems to be trivial. One of the possible reasons could be that the text-based input features used in the second task were insufficient to predict pitch accent. This insufficiency leads to that some patterns are extremely difficult to learn, which could not be remedied even by combining multiple trees. Therefore, better feature sets are needed in future studies. For example, since we predict pitch accent at syllable level, we may need to convert part-of-speech from a word-level feature to a syllable-level feature.

It is usually difficult to compare results obtained from different studies directly, because the corpus, prosodic labeling scheme, input feature set, and many other important experimental configurations could be different. Nevertheless, the present work shares many similarities with [7][8], and hence the results may be comparable. In Ross and Ostendorf[7], a dynamical system is developed to predict pitch accent using acoustic features and 84.61% (calculated from Table 1 in their paper) overall correct rate is achieved. In this work, both bagging and boosting yield 84.71% overall correct rate. In [8], decision trees combined with Markov sequence models are used to predict pitch accent using text-based features and 80.17% (calculated from Table VI in their paper) overall correct rate is obtained. Correspondingly, in the second experiment of the present study, bagging and boosting achieve 80.64% and 80.50% overall correct rate, respectively. Note that simpler feature sets were used in this work. Moreover, our system seems to be less complex and easier to implement.

It has been shown by many studies that boosting usually performs better than bagging (e.g. [3]). The results of bagging and boosting in this work, however, seem to be quite similar. During the experimentation, we noticed that bagging seems to be faster in reducing error rate. In other words, to achieve similar performance, bagging needs less iterations or fewer classifiers. Additionally, the boosting algorithm is essentially sequential, whereas bagging can be executed in parallel. Thus, to build a prosodic event recognition system, bagging seems to be a better choice to begin with. It should be noted that our boosting implementation is the simplest one for multi-class problems. We expect better results be achieved by using more sophisticated versions, such as AdaBoost.M2 [4].

4. CONCLUSIONS

In summary, we have described the application of ensemble machine learning to pitch accent prediction problem. CART, bagging with CART, and boosting with CART were evaluated under three experiment conditions: acoustic feature only; text-based feature only; both acoustic and text features. Novel acoustic features derived from underlying pitch targets were developed. In all three experiments, ensemble learning yields more favorable results than single CART. This is encouraging because it indicates that by combining multiple decision trees we can consistently improve system performance without adding much complexity. We are quite optimistic that even better results could be obtained with more sophisticated input features and ensemble learning algorithms, but those experiments remain to be done.

5. ACKNOWLEDGEMENT

This study was supported in part by NIH grant DC03902.

6. REFERENCES

[1] Breiman, L. “Bagging predictors,” *Machine Learning*, 26(2): 123-140, 1996.

[2] Conkie, A., Riccardi, G., and Rose, R. C., "Prosody recognition from speech utterances using acoustic and linguistic based models of prosodic events," *Proc. of Eurospeech*, Budapest, Hungary, pp. 523-526, 1999.

[3] Dietterich T.G. “Machine learning research: Four current directions,” *AI Magazine*, 18(4):97-136, 1999.

[4] Freund, Y. and Schapire, R.E. “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, 55(1): 119-139, 1997.

[5] Hirschberg, J. "Pitch accent in context: predicting intonational prominence from text," *Artificial Intelligence*, 63:305-340, 1993.

[6] Muller, A.F. and Hoffmann, R. “A neural network model and a hybrid approach for accent label prediction,” *Proc. of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Perthshire, Scotland, 2001.

[7] Ross, K. and Ostendorf, M., "A dynamical system model for recognising intonation patterns," *Proc. of Eurospeech*, Madrid, pp. 993–996, 1995.

[8] Ross, K. and Ostendorf, M. “Prediction of abstract prosodic labels for speech synthesis,” *Computer Speech and Language*, 10: 305-340, 1993.

[9] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J., “ToBI: A standard for labelling English prosody”, *Proc. of ICSLP*, Banff, Alberta, pp. 867-870, 1992.

[10] Sun, X. "Predicting Underlying Pitch Targets for Intonation Modeling," *Proc. of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Perthshire, Scotland, 2001.

[11] Sun, X., "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," *Proc. of ICASSP*, Orlando, Florida, 2002.

[12] Taylor, P., Black, A., and Caley, R. *Introduction to the Edinburgh Speech Tools*, 1999. http://www.cstr.ed.ac.uk/projects/speech_tools/.

[13] Xu, Y. and Wang, E., “Pitch targets and their realization: Evidence from Mandarin Chinese”, *Speech Communication* 33 (4), 319-337, 2001.

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	93.73%(1808)	17.87%(121)	38.86%(82)	91.43%(32)
High	5.29%(102)	77.55%(525)	46.45%(98)	8.57%(3)
Downstep	0.98%(19)	4.58%(31)	14.69%(31)	0%(0)
Low	0%(0)	0%(0)	0%(0)	0%(0)

Table 3: Results of pitch accent recognition using acoustic features with single CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	95.08% (1834)	15.51%(105)	37.91%(80)	88.57%(31)
High	4.35%(84)	82.42%(558)	50.71%(107)	8.57%(3)
Downstep	0.52%(10)	2.07%(14)	10.90%(23)	0%(0)
Low	0.05%(1)	0%(0)	0.47%(1)	2.86%(1)

Table 4: Results of pitch accent recognition using acoustic features with bagging CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	94.82%(1829)	13.59%(92)	31.28%(66)	85.71%(30)
High	4.56%(88)	80.35%(544)	48.82%(103)	2.86%(1)
Downstep	0.62%(12)	5.76%(39)	19.43%(41)	5.71%(2)
Low	0%(0)	0.30%(2)	0.47%(1)	5.71%(2)

Table 5: Results of pitch accent recognition using acoustic features with AdaBoost CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	90.82%(1755)	19.79%(134)	24.64%(52)	17.14%(6)
High	7.47%(144)	75.48%(511)	60.19%(127)	77.14%(27)
Downstep	1.71%(33)	4.73%(32)	15.17%(32)	5.71%(2)
Low	0%(0)	0%(0)	0%(0)	0%(0)

Table 6: Results of pitch accent prediction using text features with single CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	92.43%(1783)	24.08%(163)	26.07%(55)	20%(7)
High	6.22%(120)	71.20%(482)	57.35%(121)	71.43%(25)
Downstep	1.35%(26)	4.73%(32)	16.59%(35)	8.57%(3)
Low	0%(0)	0%(0)	0%(0)	0%(0)

Table 7: Results of pitch accent prediction using text features with bagging CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	91.45%(1783)	21.57%(163)	26.07%(55)	22.86%(7)
High	6.58%(120)	72.53%(482)	54.03%(121)	71.43%(25)
Downstep	1.97%(26)	5.91%(32)	19.43%(35)	5.71%(3)
Low	0%(0)	0%(0)	0.47%(1)	0%(0)

Table 8: Results of pitch accent prediction using text features with AdaBoost CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	94.56%(1824)	14.03%(95)	27.96%(59)	74.29%(26)
High	4.30%(83)	78.43%(531)	49.29%(104)	20%(7)
Downstep	1.09%(21)	7.39%(50)	22.27%(47)	2.86%(1)
Low	0.05%(1)	0.15%(1)	0.47%(1)	2.86%(1)

Table 9: Results of pitch accent prediction using both acoustic and text features with single CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	96.84%(1868)	11.96%(81)	28.44%(60)	85.71%(30)
High	2.85%(55)	83.90%(568)	52.13%(110)	5.71%(2)
Downstep	0.31%(6)	4.14%(28)	19.43%(41)	5.71%(2)
Low	0%(0)	0%(0)	0%(0)	2.86%(1)

Table 10: Results of pitch accent prediction using both acoustic and text features with bagging CART

Recognized	Hand-labeled			
	Unaccented	**High**	**Downstep**	**Low**
Unaccented	96.79%(1867)	9.31%(63)	24.17%(51)	85.71%(30)
High	2.54%(49)	83.46%(565)	50.24%(106)	5.71%(2)
Downstep	0.47%(9)	7.09%(48)	24.17%(51)	0%(0)
Low	0.21%(4)	0.15%(1)	1.42%(3)	8.57%(3)

Table 11: Results of pitch accent prediction using both acoustic and text features with AdaBoost CART