

Automatic Discriminative Measurement of Voice Onset Time

Morgan Sonderegger¹, Joseph Keshet²

¹University of Chicago, Chicago, IL

²Toyota Technological Institute, Chicago, IL

morgan@cs.uchicago.edu, jkeshet@ttic.edu

Abstract

We describe a discriminative algorithm for automatic VOT measurement, considered as an application of predicting structured output from speech. In contrast to previous studies which use customized rules, in our approach a function is trained on manually labeled examples, using an online algorithm to predict the burst and voicing onsets (and hence VOT). The feature set used is customized for detecting the burst and voicing onsets, and the loss function used in training is the difference between predicted and actual VOT. Applied to initial voiceless stops from two corpora, the algorithm compares favorably to previous work, and the agreement between automatic and manual measurements is near human inter-judge reliability.

Index Terms: voice onset time, SVM, discriminative prediction, structured prediction

1. Introduction

Voice onset time (VOT), the difference between the onset of a stop’s burst and the onset of voicing in the following phone, is an important perceptual cue to stop voicing and place. VOT is measured in many clinical and research studies every year, requiring hundreds of transcriber-hours; for example when studying how communication disorders affect speech [1] or how languages differ in the phonetic cues to stop contrasts [2, 3]. To-date there is no reliable method for automatically determining VOT. To replace manual measurement, automatic measurement would need precision on the order of 2–5 ms, as the burst and voicing onsets are often highly transient.

One line of previous work on automatic measurement of VOT has focused mostly on its use in practical settings: speech recognition tasks [4, 5, 6], phonetic measurement [7], or accented speech detection [8]. In these studies, except [6], manual and automatic measurements are not compared.

In this paper we are concerned with comparing manual and automatic measurements. This task has been the subject of some previous studies, each of which use sets of rules acting on observations of feature vectors to determine the burst and voicing onsets for a given stop; the particular features chosen allow for high time resolution. Das and Hansen [9] use features derived from the Teager Energy Operator representation of the signal. Stouten and van Hamme [6] use features extracted from time-frequency reassigned spectra, and Yao [10] uses MFCC spectral templates with a low frame size (1 ms).

Our approach differs from [6, 9, 10] in an important aspect. Instead of a set of customized rules to estimate VOT, we use a discriminative large-margin *learning* algorithm. The advantage of margin-based discriminative learning algorithms stems from the fact that the objective function used during the learning phase is tightly coupled with the decision task one needs to perform. Given a set of training data – that is, manually-labeled

speech segments – a function is trained to predict the VOT on unseen data. The training procedure is designed to minimize the difference between the predicted and manually-measured VOT, both on the training set and on unseen examples.

One well-known discriminative learning algorithm is the support vector machine (SVM). The classical SVM algorithm is designed for simple decision tasks, such as binary classification and regression. The task of predicting VOT is more complex: the input is a speech segment of arbitrary length, and the goal is to predict the difference in time between two acoustic events in the signal. Our proposed method is based on recent advances in kernel machines and large margin classifiers for predicting sequences [11, 12].

The VOT measurement algorithm we develop is based on mapping the speech signal and the target burst/voicing onset pair into a vector space endowed with an inner product. Our learning procedure results in a classifier in this vector space, which aims to separate the burst/voicing onset pairs corresponding to manually-labeled data from all other possible onset pairs. In this sense our method is closely related to work by Keshet et al. [13] on discriminative forced alignment of speech; here we use a different loss function and a different set of feature maps.

2. Problem Setting

In the problem of VOT measurement, we are given a segment of speech, beginning with a stop consonant (plosive) followed by a voiced phone. The goal is to predict the time difference between the onset of the stop burst and the onset of voicing in the following phone. The speech segment can be of arbitrary length, and its beginning need not be precisely synchronized with the stop’s burst or closure; it is only required that the segment begin before the burst onset. This setting supports a more rich definition of the problem as follows: given a speech utterance and an orthographic transcription, the goal is to find the duration of all VOTs. This can be achieved by utilizing a forced aligner to roughly find the location of the stops which are followed by a voiced phone and then call the procedure to find the VOTs.

We represent the speech signal by a sequence of acoustic feature vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, where each \mathbf{x}_t ($1 \leq t \leq T$) is a D -dimensional vector. We denote the domain of the feature vectors by $\mathcal{X} \subset \mathbb{R}^D$. Naturally, different signals have different lengths, and thus T is not fixed; we denote by \mathcal{X}^* the set of all finite-length sequences over \mathcal{X} . We define the *label* of $\bar{\mathbf{x}}$ as a pair of numbers: $t_b \in \mathcal{T}$, the onset of the burst (in frames), and $t_v \in \mathcal{T}$, the onset of voicing of the following phone, where $\mathcal{T} = \{1, \dots, T\}$. We assume here that $t_b < t_v$, and leave the case of “prevoiced” stops to future work. Our goal is to learn a function f from the domain of all speech segments \mathcal{X}^* to the domain of all onset pairs \mathcal{T}^2 .

3. Learning Apparatus

We follow a supervised learning approach, where the function f is learned from a training set of examples. Each example consists of a speech segment $\bar{\mathbf{x}}$ and a label (t_b, t_v) . Our goal is to find a function which performs well on the training set, as well as on unseen examples. The performance of f is measured by the percentage of predicted VOT values, $t_v - t_b$, which are within a time threshold of the manually-labeled values.

Given an example $(\bar{\mathbf{x}}, t_b, t_v)$, let $(\hat{t}_b, \hat{t}_v) = f(\bar{\mathbf{x}})$ be the predicted onset pair. The *cost* associated with predicting (\hat{t}_b, \hat{t}_v) when the manually-labeled pair is (t_b, t_v) is measured by a cost function, $\gamma : \mathcal{T}^2 \times \mathcal{T}^2 \rightarrow \mathbb{R}$. The function used in our experiments is of the form:

$$\gamma((t_b, t_v), (\hat{t}_b, \hat{t}_v)) = \max\{(|\hat{t}_v - \hat{t}_b| - (t_v - t_b)) - \epsilon, 0\}, \quad (1)$$

that is, the only VOT differences greater than a threshold ϵ are penalized. This cost function takes into account that manual measurements are not in general exact; ϵ can be adjusted according to the level of measurement uncertainty in a dataset. For brevity, we denote $\gamma = \gamma((t_b, t_v), (\hat{t}_b, \hat{t}_v))$.

Following the structured prediction scheme, we make use of a predefined set of N feature maps, $\{\phi_j\}_{j=1}^N$, each a function of the form $\phi_j : \mathcal{X}^* \times \mathcal{T}^2 \rightarrow \mathbb{R}$. That is, each feature map takes a speech segment $\bar{\mathbf{x}}$ and a proposed onset pair (t_b, t_v) , and returns a scalar which, intuitively, represents the confidence in the suggested pair, and should be high when (\hat{t}_b, \hat{t}_v) is close (measured by γ) to the manually-labeled pair (t_b, t_v) . We restrict ourselves to the set of linear functions of the feature maps,

$$f(\bar{\mathbf{x}}) = \arg \max_{(t_b, t_v)} \mathbf{w} \cdot \phi(\bar{\mathbf{x}}, t_b, t_v), \quad (2)$$

where $\mathbf{w} \in \mathbb{R}^N$ is a vector of weights, denoting the relative importance of the feature maps, that we need to learn. In § 3.1 we describe the set of feature maps we used, and in § 3.2 we describe how \mathbf{w} is estimated from a training set of examples.

3.1. Features and Feature Maps

Consider the speech segment $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ consisting of T frames, where each acoustic feature vector \mathbf{x}_t consists of D features. We extracted 7 ($D = 7$) acoustic features every 1 ms. The first 4 features refer to an STFT taken with a 5 ms Hamming window: the total spectral energy, E_{total} ; the energy between 50–1000 Hz, E_{low} ; the energy above 3000 Hz, E_{high} ; and the *Wiener entropy*, H_{wiener} , a measure of spectral flatness:

$$H_{\text{wiener}}(t) = \log \int |P(f, t)|^2 df - \int \log |P(f, t)|^2 df,$$

where $P(f, t)$ is the STFT of the signal at frequency f and time t . The low frame rate and window size are used for fine time resolution, because the burst and voicing onsets are highly transient events.

The fifth feature, R_l , is extracted from the signal itself: the maximum of the FFT of its autocorrelation function, starting 6 ms before and ending 18 ms after the frame center. The sixth feature is the pitch track given by the Sha & Saul pitch tracker [14]; the seventh feature, V , is the 0/1 output of a voicing detector based on the RAPT pitch tracker [15]. Features 6–7 are smoothed with a 5 ms Hamming window.

Before presenting the feature maps, we introduce notation for *local differences*. Let x^d be the d -th acoustic feature ($1 \leq d \leq D$). Let $\Delta_t^s(x^d)$ be the local difference of resolution s applied to the acoustic feature x^d , defined as the difference

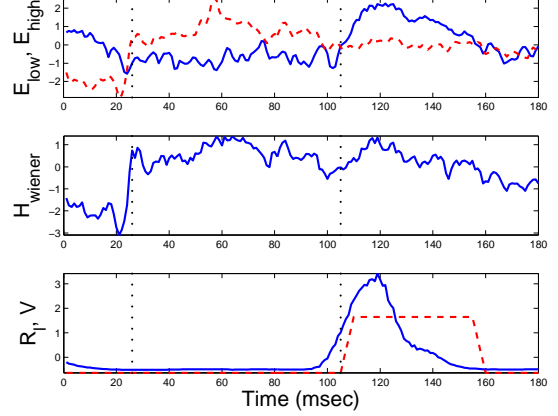


Figure 1: Acoustic features for one example (“can’t”). The vertical lines correspond to the burst and voicing onsets. H_{wiener} and E_{high} rapidly increase at the burst onset, while E_{low} , R_l , and V rapidly increase at the voicing onset.

between (1) the mean of x^d over frames $\{t, \dots, \min(t+s, T)\}$ and (2) the mean of x^d over frames $\{\max(t-s, 0), \dots, t\}$. This quantity provides a local approximation of the derivative of x^d at frame t , with resolution parametrized by s .

We now turn to the feature maps. For each example $(\bar{\mathbf{x}}, t_b, t_v)$, 59 feature maps ($N = 59$) were calculated:

- $\log E_{\text{total}}(t_b)$, $\log E_{\text{high}}(t_b)$, $H_{\text{wiener}}(t_b)$ (3 functions)
- $\log E_{\text{total}}(t_v)$, $\log E_{\text{low}}(t_v)$, $\log E_{\text{high}}(t_v)$, $H_{\text{wiener}}(t_v)$, $R_l(t_v)$ (5 functions)
- $\Delta_t^s(x^d)$ for $s \in \{5, 10, 15\}$, $t \in \{t_b, t_v\}$, and $d \in \{1, \dots, 5\}$. (30 functions)
- $\Delta_t^s(x^d)$ for $s \in \{5, 10, 15\}$, $t = t_v$ and features $d \in \{6, 7\}$ (6 functions)
- For $s \in \{5, 10\}$ (8 functions):
 - mean and maximum of $\Delta_t^s(R_l)$ over $t \in [t_b, t_v]$.
 - mean and maximum of $\Delta_t^s(R_l)$ over $t \in [t_b, \min(t_v - 10, t_b)]$.
- For the features $x^d \in \{H_{\text{wiener}}, \log E_{\text{high}}\}$ (4 functions):
 - mean of x^d over $[t_b, t_v]$ minus the mean over $[1, t_b]$
 - maximum of x^d over $[1, \max(t_b - 5, 1)]$
- For the features $x^d \in \{H_{\text{wiener}}, \log E_{\text{high}}, V\}$, the mean of x^d over $[1, \max(t_b - 5, 1)]$. (3 functions)

Intuitively, we expect t_b and t_v to occur at points of rapid change, where some local difference features spike. As in the example in Fig. 1, there is often a rapid increase in H_{wiener} and E_{high} at the burst onset t_b , and a rapid increase in R_l , E_{low} , and V at the voice onset t_v .

3.2. A Discriminative Algorithm

We now describe a simple iterative algorithm for learning the weight vector \mathbf{w} , based on the family of algorithms described in [16] for structured prediction. The algorithm receives as input a training set $S = \{(\bar{\mathbf{x}}^i, t_b^i, t_v^i)\}_{i=1}^m$ of examples and a parameter C , and works in rounds. At each round, an example is presented to the algorithm, and \mathbf{w} is updated. We denote by \mathbf{w}^i the value of the weight vector after the i -th iteration. Initially we set $\mathbf{w}^0 = \mathbf{0}$. Let $(\hat{t}_b^i, \hat{t}_v^i)$ be the predicted onset pair for the i -th example according to \mathbf{w}^{i-1} ,

$$(\hat{t}_b^i, \hat{t}_v^i) = \arg \max_{(t_b, t_v)} \mathbf{w}^{i-1} \cdot \phi(\bar{\mathbf{x}}^i, t_b, t_v). \quad (3)$$

We set the weight vector \mathbf{w}^i to be the minimizer of the following optimization problem,

$$\min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} \|\mathbf{w} - \mathbf{w}^{i-1}\|^2 + C\xi \quad (4)$$

$$\text{s.t. } \mathbf{w} \cdot \phi(\bar{\mathbf{x}}^i, t_b^i, t_v^i) - \mathbf{w} \cdot \phi(\bar{\mathbf{x}}^i, \hat{t}_b^i, \hat{t}_v^i) \geq \gamma - \xi,$$

where C serves as a complexity-accuracy trade-off parameter as in the SVM algorithm and ξ is a non-negative slack variable, which indicates the loss of the i -th example. Intuitively, we would like to minimize the loss of the current example (the slack variable ξ) while keeping the weight vector \mathbf{w} as close as possible to our previous weight vector \mathbf{w}^{i-1} . The constraint makes the projection of the manually-labeled onset pair (t_b^i, t_v^i) onto \mathbf{w} higher than the projection of the predicted pair $(\hat{t}_b^i, \hat{t}_v^i)$ onto \mathbf{w} by at least the cost function between them. It can be shown (see [16]) that the solution to the above optimization problem is

$$\mathbf{w}^i = \mathbf{w}^{i-1} + \alpha^i \Delta\phi^i, \quad (5)$$

where $\Delta\phi^i = \phi(\bar{\mathbf{x}}^i, t_b^i, t_v^i) - \phi(\bar{\mathbf{x}}^i, \hat{t}_b^i, \hat{t}_v^i)$. The value of the scalar α^i is based on the cost function γ , the different scores that the manually-labeled onset pair and the predicted pair received according to \mathbf{w}^{i-1} , and a parameter C . Formally,

$$\alpha^i = \min \left\{ C, \max \{ \gamma - \mathbf{w}^{i-1} \cdot \Delta\phi^i, 0 \} / \|\Delta\phi^i\|^2 \right\}. \quad (6)$$

Given a training set of m examples we iterate over its elements, possibly M epochs, and update the weight vector $M \cdot m$ times. A common procedure is to use the latest weight vector \mathbf{w}^{Mm} to classify unseen utterances. Another alternative, often resulted with much better performance is to use the average of $\{\mathbf{w}^1, \dots, \mathbf{w}^{Mm}\}$ rather than just its last element. We denote this average by \mathbf{w}^* . A theoretical analysis shows that with i.i.d. assumptions over the data, the average estimate is optimal, in the sense that, with high probability the loss suffered over new speech segment will be small [17].

4. Data

The data are audio of English words beginning with initial voiceless stops ($/p/, /t/, /k/$), drawn from two datasets.

TIMIT: We considered all words (excluding SA1 and SA2 utterances) transcribed as beginning with an unvoiced stop closure and burst, followed by a voiced segment, resulting in 4126 words from all 630 speakers. The VOT boundaries and word boundaries were determined manually or automatically for different subsets of the data, as described below.

Big Brother (BB): A corpus of speech from Big Brother UK, a reality television show [18]. 704 word-initial voiceless stops, have been manually annotated for VOT by (one of) two transcribers (85%/15%); the end of each word has also been annotated. Words come from spontaneous speech in the “diary room”, an acoustically clean environment. Data come from 4 speakers, wearing individual microphones. Because the *beginnings* of words have not been annotated, we took each word to begin 25 ms before its burst. Differently to the TIMIT data, we did not exclude stops with no preceding closure. Stops with no following voiced segment were kept if there was still abrupt spectral change at the end of the burst, and excluded otherwise.

5. Experiments

To evaluate our algorithm, we performed experiments using the TIMIT and BB datasets. In all experiments, we only considered

burst onsets t_b within 0–150 ms of the start of the word, and voicing onsets t_v 15–200 ms later than t_b ; this step attempts to restrict the algorithm’s focus to the first two segments of each word. We fix $C = 5$, as varying it has little effect on performance, and we set $\epsilon = 4$ in all experiments.

5.1. Big Brother

Because of the relatively small amount of BB data, we used four-fold cross-validation. For each speaker, our algorithm was applied to the other three speakers’ data, with $M = 3$. The resulting weight vector \mathbf{w}^* was used to determine (t_b, t_v) , and hence VOT, for that speaker’s stops.

To evaluate the algorithm’s performance, we ask how the difference between the automatic and manual measurements compare to differences between human transcribers. For a subset of the data (65 stops), VOT was measured by a second transcriber. Fig. 2 shows the distribution across stops of the difference between the automatic and manual measurements, as well as the distribution of inter-transcriber differences. The algorithm performs very well: automatic/manual agreement is *higher* than inter-transcriber agreement.

5.2. TIMIT

For the TIMIT data, we trained a single weight vector \mathbf{w}^* , then applied it to several test sets.

For training, we applied our algorithm to all examples from the TIMIT training set (3088 stops), with $M = 2$. For each example, the word boundaries were taken from the TIMIT transcription, and VOT boundaries (t_b, t_v) were taken to be the burst boundaries. Because the burst sometimes ends after the onset of voicing, this step is an approximation, one which allows us to take advantage of the size of TIMIT.

For testing, we first applied \mathbf{w}^* to the core TIMIT test set, and to the complete test set. To check how much the automatic measurement process depends on the accuracy of word boundaries, we ran experiments both using the TIMIT boundaries, and boundaries given by forced alignment (FA), carried out as follows. For each utterance, the orthographic transcription was converted to phones using the TIMIT dictionary, with `sil` added before each closure; alignment was then performed as in [13]. For some examples (5.4%/7.13% for core/complete), FA did not give a left boundary beginning 0–150 ms before the burst beginning; these examples were not used in testing. Running time was 10 minutes for the complete test set (75%/25% train/test) on an Intel Xeon 2.66 GHz, running Linux.

Fig. 2 shows the distribution of differences between automatic and manual VOTs (taken to be the burst duration, as in training), for experiments using both test sets and both types of word boundaries. Using FA word boundaries decreases performance slightly over the core test set, compared to manual word boundaries, but has little impact over the complete test set.

5.3. Comparison with previous work

Our results can be most closely compared with Stouten & van Hamme (SvH; [6]), who automatically measured VOT for stops from TIMIT, using a knowledge-based algorithm (see §1). SvH consider voiced and voiceless stops, in all positions. They took manual measurements for a subset of 582 stops (the “manual” dataset), and compared these to their automatic measurements.

We applied our algorithm to the 293 voiceless stops from SvH’s “manual” dataset, once by training \mathbf{w}^* on all BB data (with $M = 1$) and once using \mathbf{w}^* from the TIMIT experiments

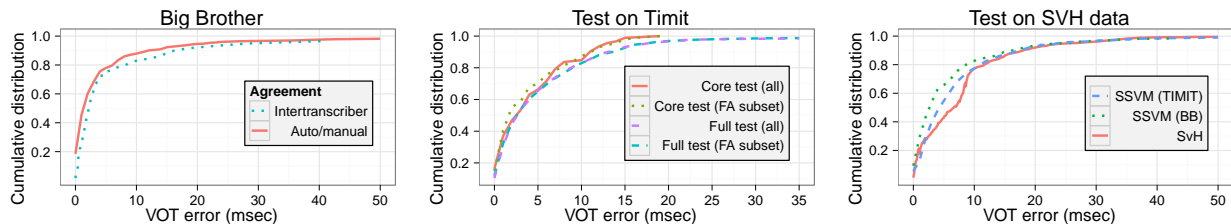


Figure 2: Cumulative distributions of absolute differences between manual and automatic VOT measurements, in Big Brother (§5.1), TIMIT (§5.2), and Stouten & van Hamme (§5.3) datasets. Details in text.

just described, yielding two sets of automatic VOT measurements. Because we are now not only dealing with stops in initial position, the left “word” boundary for each example was determined differently from above. Each example was taken to start at the beginning of the segment preceding the burst (the closure, if one was present), and end at the right word boundary; all boundaries were taken from TIMIT.

Fig. 2 shows the distribution of difference in VOT, t_{diff} , relative to SvH’s manual measurements, for the three automatic measurement methods: training on BB, training on TIMIT, and SvH¹. Using either \mathbf{w}^* , our algorithm arguably performs as well as SvH for examples with t_{diff} greater than 15 ms, and better than SvH for examples with t_{diff} less than 15 ms. We note that our algorithm was only trained on initial stops, but tested on stops in all positions, and the training and testing data were either from different datasets (BB vs. TIMIT) or labeled by different annotators (SvH vs. TIMIT annotators). This might show the robustness of the algorithm to a different environments.

Table 1: Experimental performance using metrics from [9, 10].

		RMS (ms)	$\geq 10\%$
TIMIT complete	all	8.66	0.35
	FA	7.56	0.36
TIMIT core	all	5.28	0.34
	FA	5.18	0.31
Big Brother		7.74	0.23
Yao [10]		10.8	–
Das/Hansen [9]		–	0.25

Our results can be compared less directly with other studies where automatic and manual measurements are compared. Das & Hansen [9] report the percentage of stops where automatic and manual measurements differ by $\geq 10\%$; Yao [10] reports RMS error for automatic measurement of the burst onset only. Table 1 gives these metrics for our TIMIT and BB experiments. Though comparison is difficult because of the different datasets used, all experiments’ RMS error out-performs [10], and our best-performing experiment (BB) outperforms [9]. We note that Das & Hansen consider isolated words, where VOTs are much easier to measure than in the types of speech considered here (read or conversational).

We would ideally compare the results of discriminative measurement with the “gold standard” of human interjudge reliability (IJR), but are not aware of recent studies giving IJR measures for the types of speech considered here. However, we have shown for the BB dataset that the agreement between our automatic and manual measurements is *better* than between two human transcribers, and tentatively conclude that our algorithm achieves near-IJR performance.

Acknowledgments. We thank Max Bane and Ed King for help with VOT data, and Channel 4/Endemol for permission to use the Big Brother data.

¹We thank Hugo van Hamme for providing the manually-annotated data. The error distribution is taken from [6, Fig. 5] (voiceless only).

6. References

- [1] P. Auzou, C. Ozsancak, R. Morris, M. Jan, F. Eustache, and D. Hannequin, “Voice onset time in aphasia, apraxia of speech and dysarthria: a review,” *Clin. Linguist. Phonet.*, vol. 14, no. 2, pp. 131–150, 2000.
- [2] L. Lisker and A. Abramson, “A cross-language study of voicing in initial stops: acoustical measurements,” *Word*, vol. 20, pp. 384–422, 1964.
- [3] T. Cho and P. Ladefoged, “Variation and universals in VOT: evidence from 18 languages,” *J. Phonetics*, vol. 27, no. 2, pp. 207–229, 1999.
- [4] A. Ali, “Auditory-based acoustic-phonetic signal processing for robust continuous speech recognition,” Ph.D. dissertation, University of Pennsylvania, 1999.
- [5] P. Niyogi and P. Ramesh, “The voicing feature for stop consonants: Recognition experiments with continuously spoken alphabets,” *Speech Commun.*, vol. 41, no. 2-3, pp. 349–367, 2003.
- [6] V. Stouten and H. van Hamme, “Automatic voice onset time estimation from reassignment spectra,” *Speech Commun.*, vol. 51, no. 12, pp. 1194–1205, 2009.
- [7] C. Fowler, V. Sramko, D. Ostry, S. Rowland, and P. Hallé, “Cross language phonetic influences on the speech of French-English bilinguals,” *J. Phonetics*, vol. 36, no. 4, pp. 649–663, 2008.
- [8] A. Kazemzadeh, J. Tepperman, J. Silva, H. You, S. Lee, A. Alwan, and S. Narayanan, “Automatic detection of voice onset time contrasts for use in pronunciation assessment,” in *INTERSPEECH*, 2006.
- [9] S. Das and J. Hansen, “Detection of Voice Onset Time for unvoiced stops using the Teager Energy Operator for automatic detection of accented English,” in *NORSIG*, 2004.
- [10] Y. Yao, “Closure duration and VOT of word-initial voiceless plosives in English in spontaneous connected speech,” *UC Berkeley Phonology Lab Annual Report*, pp. 183–225, 2007.
- [11] B. Taskar, C. Guestrin, and D. Koller, “Max-margin Markov networks,” in *NIPS*, 2003.
- [12] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, “Support vector machine learning for interdependent and structured output spaces,” in *ICML*, 2004.
- [13] J. Keshet, S. Shalev-Shwartz, Y. Singer, and D. Chazan, “A large margin algorithm for speech-to-phoneme and music-to-score alignment,” *IEEE T. Audio Speech*, vol. 15, no. 8, pp. 2373–2382, 2007.
- [14] F. Sha and L. Saul, “Real-time pitch determination of one or more voices by nonnegative matrix factorization,” *NIPS*, 2005.
- [15] D. Talkin, “A robust algorithm for pitch tracking (RAPT),” in *Speech coding and synthesis*, W. Kleijn and K. Paliwal, Eds. Elsevier, 1995, pp. 495–518.
- [16] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, 2006.
- [17] N. Cesa-Bianchi, A. Conconi, and C. Gentile, “On the generalization ability of on-line learning algorithms,” *IEEE T. Inform. Theory*, vol. 50, no. 9, pp. 2050–2057, 2004.
- [18] M. Bane, P. Graff, and M. Sonderegger, “Longitudinal phonetic variation in a closed system,” in *Chicago Ling. Soc.* 46, to appear.