Graphical Model Approach to Pitch Tracking

Xiao Li, Jonathan Malkin and Jeff Bilmes

Department of Electrical Engineering
University of Washington, Seattle
{lixiao,jsm,bilmes}@ee.washington.edu

Abstract

Many pitch trackers based on dynamic programming require meticulous design of local cost and transition cost functions. The forms of these functions are often empirically determined and their parameters are tuned accordingly. Parameter tuning usually requires great effort without a guarantee of optimal performance. This work presents a graphical model framework to automatically optimize pitch tracking parameters in the maximum likelihood sense. Therein, probabilistic dependencies between pitch, pitch transition and acoustical observations are expressed using the language of graphical models, and probabilistic inference is accomplished using the Graphical Model Toolkit (GMTK). Experiments show that this framework not only expedites the design of a pitch tracker, but also yields remarkably good performance for both pitch estimation and voicing decision.

1. Introduction

Pitch tracking has drawn increased attention in speech coding, synthesis and recognition and in prosody modeling. In the "Vocal Joystick" project recently funded by NSF and carried out at the University of Washington, pitch (along with formants, power and other features) is intended to be utilized as an acoustic feature for human-machine interaction. For example, the pitch trajectory can serve as one coordinate for 3-D continuous motion control. Therefore, a robust pitch tracking system is of significant importance and interest.

Many state-of-the-art pitch trackers resemble the methodology proposed by [1], which consists of three steps: preprocessing, pitch candidate generation, and post-processing by dynamic programming (DP). The first step involves signal conditioning techniques, and the second step selects pitch candidates and computes their "scores" by applying certain pitch detection algorithms (PDA) to the local frame acoustics. In the post-processing step, the cost $C_{t,j}$ of proposing pitch candidate j at frame t is computed as follows,

$$C_{t,j} = f_t^{local}(j) + \min_i \{C_{t-1,i} + f_{t-1,t}^{tran}(i,j)\} \qquad (1)$$

where the local cost function f^{local} takes into account the scores obtained from the second step, and the transition cost function f^{tran} models the penalty of transitioning from candidate i of the previous frame to candidate j of the current frame.

The forms of these cost functions are usually empirically determined and their parameters are often tuned by algorithms such as gradient descent [1]. This process, however, remains a difficult problem and a time-consuming task. First, f^{local} has to be optimized each time a different PDA is applied. For example, PDAs can be designed in several domains including time, spectral, cepstral and their combinations [2]. While classic PDAs like the normalized cross-correlation function (NCCF) [3] are popularly used, new algorithms such as ACOLS [4], JTFA [5] and YIN [6], are increasingly coming into play. In order to evaluate, compare and eventually implement these techniques, a large amount of time has to be spent deciding the form of f^{local} and tuning its parameters. Second, ideally f^{tran} should be adapted when the pitch tracker is exposed to another language, or applied to another application. This is because different languages and applications may follow very different pitch transition patterns. For example, in the Vocal Joystick project, pitch is allowed to change arbitrarily, unlike the "gradually changing" assumptions made by most pitch tracking systems adapted to natural speech. Therefore, the local and transition cost functions optimized for certain PDAs and applications may not be the most appropriate for others.

The goal of this work is to expedite the design of new pitch trackers customized for specific applications. Extending the basic idea of [7], this work provides a graphical model framework to learn a pitch tracker from data. Therein, a PDA or a pitch transition pattern can be easily incorporated into the system with parameters automatically estimated using statistical methods. Furthermore, since the parameters are optimized in the maximum likelihood sense, both pitch estimation and voicing decision give better performance. The Graphical Model Toolkit (GMTK) [8] was utilized to implement our framework (GMTK is a publically available toolkit for developing graphical model and dynamic Bayesian network based speech, language, and time-series analysis systems).

The rest of the paper is organized as follows: Section 2 describes our graphical models for learning and decoding and discusses practical issues associated with parameter estimation. Section 3 presents experiments and results, followed by discussion in the last section.

2. Graphical Model Framework

2.1. Graph structure and local probability models

Graphical models are a flexible, concise, and expressive probabilistic modeling framework with which one may rapidly specify a vast collection of statistical models. Our graphical model framework for pitch tracking (decoding) is depicted in Figure 1. The shaded nodes represent variables observed at decode time, whereas the unshaded nodes are hidden.

1. The random variable Q_t is discrete with cardinality N, corresponding to $N-1$ possible pitch periods (voiced states) plus one unvoiced state (with index N) at frame t. Q_t has no

This work was funded by NSF under Award ISS/ITR-0326382

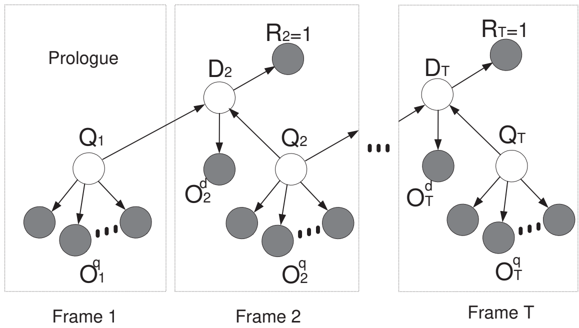

Figure 1: Decoding graph

parents, but has a prior distribution $\pi_i^q \triangleq P(Q_t = i)$.

2. The random variable D_t is also discrete with cardinality M, corresponding to M transition patterns coarsely quantized from N^2 possible (Q_{t-1}, Q_t) pairs. The dependency between Q_{t-1}, Q_t and D_t is represented by a deterministic table. Specifically, the set $\{1..N\} \times \{1..N\}$ is partitioned into M non-overlapping subsets $\mathcal{S}_m$, $m = 1..M$, and $P(D_t = m|Q_{t-1} = i, Q_t = j) = 1$ iff $(i, j) \in \mathcal{S}_m$. In this work, we initially use a simple partition scheme:

$$\begin{array}{rcl} \mathcal{S}_1 & = & \{(i,j) : i = N, j \neq N\}; \\ \mathcal{S}_2 & = & \{(i,j) : i \neq N, j = N\}; \\ \mathcal{S}_m & = & \{(i,j) : L_m \leq j - i < U_m\}; \quad m = 3..M, \end{array} \tag{2}$$

where L_m and U_m are (respectively) lower and upper bounds evenly spaced at integers between $-N+2$ and $N-2$. In other words, $\mathcal{S}_1$ corresponds to unvoiced-to-voiced transitions; $\mathcal{S}_2$ corresponds to voiced-to-unvoiced transitions; voiced pitch transitions are clustered into $M-2$ patterns based on pitch period difference; and the unvoiced-to-unvoiced transition belong to the same subset as the voiced pitch transition where $i = j$.

3. There is a dummy binary node R_t parented by D_t with conditional probability

$$\pi_m^d \triangleq P(R_t = 1|D_t = m) = \frac{\sum_{(i,j)\in\mathcal{S}_m} C(i,j)}{\sum_{m=1}^{M}\sum_{(i,j)\in\mathcal{S}_m} C(i,j)}, \tag{3}$$

where $C(i, j)$ is the count function of the instances of the event $\{Q_{t-1} = i, Q_t = j\}$ in the training data. The purpose of this dummy node is to provide soft evidence for D_t, and this evidence is encoded using the histogram of the M pitch transition patterns. Note that for the purposes of inference and decoding, the results would be identical with a π_m^d multiplied by any positive scalar. We keep this expression of soft evidence, as it is amenable to standard smoothing methods (see Section 2.4).

4. The children of Q_t and D_t are continuous observations O_t^q and O_t^d. They are both obtained from acoustic signals, which will be discussed in the next subsection.

Figure 1 is a valid graph for pitch tracking when considered in accordance to Bayesian network semantics. It captures dependency between pitch and local acoustics, and that between pitch transition pattern and acoustical changes. Also, by modeling Q_{t-1} and Q_t as parents of D_t and adding dummy nodes R_t, the prior probabilities of pitch and pitch transition are simultaneously modeled in the graph, which would otherwise be hard to accomplish. Note that in our model, it is *not* the case that Q_t is independent of Q_{t+1} due to the evidence $R_t = 1, \quad \forall t$. Moreover, Figure 1 is an intuitive and efficient graph compared to its alternatives. For example, a condensed, HMM-like graph could be used instead, where Q_{t-1}, Q_t and D_t are bundled into a large single hidden node. However, this hidden node would have high cardinality, and the local probabilities in their factored form would not be straightforward either to represent or to learn.

2.2. Observation features

The observation features are crucial to the success of pitch tracking. Autocorrelation coefficients or their extended forms [3, 4, 6] can be directly used as O_t^q, corresponding to time-domain PDAs. For example, in the case of NCCF, we let $\mathbf{x_t} = (x_{1,t}, x_{2,t}, \ldots, x_{N,t})$ where $x_{i,t}$, $i = 1..N-1$, is the NCCF coefficient of the i^{th} candidate pitch period, and $x_{N,t} = \max_{i=1..N-1} x_{i,t}$. If frame t is voiced and the i^{th} candidate is the truth, $x_{i,t}$ is likely to be high (close to one ideally). On the other hand, if frame t is unvoiced, $x_{N,t}$ is likely to be low. Therefore, we model the observation distribution as

$$P(O_t^q = \mathbf{x_t}|Q_t = i) = \left\{ \begin{array}{ll} \mathcal{N}(x_{i,t}; 1, \beta^2) & i = 1..N-1 \\ \mathcal{N}(x_{i,t}; \mu, \gamma^2) & i = N \end{array} \right. \tag{4}$$

where μ is fixed at the minimum value of $x_{N,t}$ for all data. Note that these two means, 1 and μ, are set in advance and are fixed during the training of the other parameters. Since $x_{i,t} < 1$, $i = 1..N$, a high $x_{i,t}$ will lead to a high observation probability for state i. Similarly since $x_{N,t} > \mu$, a low $x_{N,t}$ will imply a high observation probability for state N, meaning frame t is likely to be unvoiced.

In this work, we choose NCCF based features in order to compare with [1] which uses NCCF coefficients in the DP. As we will see later, our graphical model automatically optimizes the parameters and significantly improves the estimation rate. We can certainly choose other features (PDAs) such as normalized YIN [6] coefficients for further improvement.

The observation features O_t^d is the power change from frame $t-1$ to t. The choice of this feature is based on the empirical observation (justified by our experiments) that there is a correlation between the change in pitch state and the change in power. For example, an utterance with decreasing pitch tends to have decreasing power, and an unvoiced-to-voiced transition tends to have increasing power. Therefore, computing the power change may help in deciding the transition patterns of the pitch and thereby reduce the estimation error rate. The corresponding observation distribution is modeled as

$$P(O_t^d = y_t|D_t = m) = \mathcal{N}(y_t; \rho_m, \sigma^2), \tag{5}$$

where y_t is the relative power change between two consecutive frames, ρ_m is the mean of the Gaussian of the m^{th} transition pattern, and σ^2 is shared by all Gaussians.

2.3. Parameter estimation and decoding

Slightly different from the decoding graph, the training graph is depicted in Figure 2. In fact, D_t is implicitly observed since it is deterministic given Q_{t-1} and Q_t. This graph makes the following conditional independence assumptions: (a) O_t^q is independent of all other variables given Q_t, as is O_t^d given D_t; (b) Q_t and D_t are independent of $Q_{1:t-2}$ and $D_{2:t-1}$, given Q_{t-1}. These statements imply that the likelihood can be de-

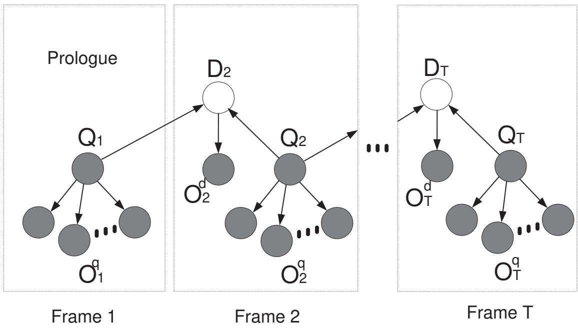

Figure 2: Training graph

composed into several factors each of which can be maximized during training:

$$
\begin{aligned}
& \ln P(Q_{1:T}, O^q_{1:T}, O^d_{2:T}) \\
= \; & \ln \sum_{d_{2:T}} P(Q_{1:T}, D_{2:T} = d_{2:T}, O^q_{1:T}, O^d_{2:T}) \\
= \; & \sum_{t=1}^{T} \ln P(Q_t = q_t) + \sum_{t=1}^{T} \ln P(O^q_t = \mathbf{x_t} | Q_t = q_t) \\
& + \sum_{t=2}^{T} \ln[\sum_m P(O^d_t = y_t | D_t = m) \\
& \cdot P(D_t = m | Q_{t-1} = q_{t-1}, Q_t = q_t)]
\end{aligned} \tag{6}
$$

Recall that $P(D_t = m|Q_{t-1} = q_{t-1}, Q_t = q_t)$ is in fact an indicator function which equals one when $(q_{t-1}, q_t) \in \mathcal{S}_m$. Therefore, only one pitch transition pattern can survive the summation over m. Plugging Equation (4) and Equation (5) into Equation (6) and taking derivatives with respect to the parameters, we can get the maximum likelihood estimation of π^q_i, β^2, γ^2, ρ_i and σ^2. Furthermore, π^d_m in Equation (3) are also easily estimated during training by computing the histogram of pitch transition patterns.

For decoding, we use the graph in Figure 1. Let $\alpha_j(t) = P(Q_t = j, O^q_{1:t}, O^d_{2:t}, R_{2:t})$. Decoding can then be written as,

$$
\begin{aligned}
\alpha_j(1) \; & = \pi^q_j P(O^q_1 = \mathbf{x_1} | Q_1 = j); \\
\alpha_j(t) \; & = \pi^q_j P(O^q_t = \mathbf{x_t} | Q_t = j) \cdot \\
& \max_i \, [\sum_m \pi^d_m P(O^d_t = y_t | D_t = m) \cdot \\
& P(D_t = m | Q_{t-1} = i, Q_t = j) \alpha_i(t-1))]
\end{aligned} \tag{7}
$$

Again, $P(D_t = m|Q_{t-1} = i, Q_t = j)$ is an indicator function. If we let $C_{t,i} = -\ln \alpha_i(t)$, Equation (7) is equivalent to the DP in Equation (1), where $K \triangleq \frac{1}{2} \ln 2\pi$, and

$$
\begin{aligned}
f^{local}_t(j) &= \begin{cases} -\ln \pi^q_j + K + \ln \beta + \frac{(1-x_{j,t})^2}{2\beta^2}, & j = 1..N-1 \\ -\ln \pi^q_j + K + \ln \gamma + \frac{(x_{j,t}-\mu)^2}{2\gamma^2}, & j = N \end{cases} \\
f^{tran}_{t-1,t}(i,j) &= -\ln \pi^d_m + K + \ln \sigma + \frac{(y_t - \rho_m)^2}{2\sigma^2}, \;\; (i,j) \in \mathcal{S}_m
\end{aligned} \tag{8}
$$

The best pitch period sequence can be obtained via backtracking after the DP terminates. The Gaussian assumption of local probabilities lead to a quadratic form of these cost functions. With parameters optimized in the maximum likelihood sense, these functions give remarkably good performance as we will see in Section 3. It is worth noting that these cost functions can take on other forms under a different distribution assumption, and the parameters can be efficiently estimated as long as good sufficient statistics exist for that distribution.

2.4. Smoothing

One issue associated with parameter learning using graphical models is the lack of training data. Certain pitch values or pitch transitions may not exist in the training set. To compensate for this problem, we smooth the priors using a method similar to Laplace smoothing [9]. In the case of the transition values,

$$\pi^d_m(\text{new}) = \frac{\pi^d_m + \lambda}{1 + M\lambda}. \tag{9}$$

In the case of pitch priors, the unvoiced state is treated separately,

$$\pi^q_i(\text{new}) = \begin{cases} \frac{N-1}{N} \frac{\pi^q_i/(1-\pi^q_N)+\lambda}{1+(N-1)\lambda} & i = 1..N-1 \\ \frac{1}{N} & i = N \end{cases} \tag{10}$$

The choice of λ depends on the amount of training data available. The transition priors can be well estimated with only a small amount of data, but the pitch priors are usually biased due to the limited number of different speakers in the training set. In practice, we often choose a small λ for π^d_m, so that $\pi^d_m(\text{new}) \approx \pi^d_m$, and choose a very large λ for π^q_i, so that $\pi^q_i(\text{new})$ is close to a uniform distribution.

3. Evaluation

3.1. Setup

Two databases were combined to create train and test sets for our graphical-model based pitch tracker. One is "Mocha-TIMIT," [10] developed at Queen Margaret University College; the other was developed at the Hong Kong University of Science and Technology for tone-estimation research.

A total of 1192 continuous English speech utterances from two male and two female speakers were allocated to the training set. The test set was comprised of 4 subsets, corresponding to the same four speakers, amounting to 647 utterances different from the train set. Laryngograph waveforms are available for all data. To obtain the pitch ground truth, we first filtered out the humming noise (the noise generated by the electronic devices) in the laryngograph, then applied ESPS pitch tracking tool "get_f0" [1] to these waveforms.

3.2. Experiments and Results

Our front-end for observation feature extraction consisted of a sequence of signal processing modules. The speech waveforms were sampled at 8kHz, and a frame was created every 10ms with a length of 40ms. Center clipping was used to remove background musical noise. NCCF coefficients of 144 possible pitch periods (50Hz–500Hz) as well as the relative power change were computed for each frame, corresponding to the observation features O^q_t and O^d_t respectively. Pitch transitions were quantized to $M = 145$ different patterns, which gave the best performance compared to several other quantization resolutions.

Our training and decoding was implemented using a new fast version of GMTK. We fully exploited GMTK's ability to arbitrarily tie and/or hold fixed portions of Gaussian and other

speaker	female 1	female 2	male 1	male 2
get_f0	5.83	2.11	3.73	1.51
GM	3.38	1.55	1.17	0.86

Table 1: % Pitch estimation GER

speaker	female 1	female 2	male 1	male 2
get_f0	13.07	12.92	25.66	12.84
GM	9.12	12.59	24.37	5.94

Table 2: % Voicing decision error rate

parameters, and also its unity-score observation distribution capabilities. The graph in Figure 2 was used for training and Figure 1 for decoding. The prior probabilities were smoothed using Equation (9) and Equation (10).

We ran both get_f0 and our graphical-model based pitch tracker on the speech waveforms of the test set, and compared the results with the ground truth generated by get_f0 from the laryngograph. The pitch trackers were evaluated in two aspects: pitch estimation and voicing decision [11]. Pitch estimation error rate is measured in terms of "gross error rate" (GER), which is the percentage of pitch estimates that deviate from the ground truth by a certain amount (20% in our experiments). The voicing decision is measured in terms of the percentage of both unvoiced-to-voiced and voiced-to-unvoiced errors. As is shown in Table 1 and Table 2, both pitch estimation GERs and voicing detection error for rates our pitch tracker were lower than those of get_f0 for all four speakers.

4. Conclusion and Discussion

In this paper, we introduce a new approach to pitch tracking whereby the pitch extractor mechanism is represented using a graphical model and is implemented using GMTK [8]. Our approach has a number of advantages owing to the fact that, given supervisory pitch extraction information, we can train the dynamic programming cost functions in a statistically grounded fashion, namely maximum likelihood. This current work does not aim to compete with existing state-of-the-art pitch tracking systems, but instead to provide a statistical framework that can optimize the performance of a pitch tracker for specific applications. With parameters learned from data, our graphical-model-based pitch tracker is able to automatically accomodate to the conditions of the training set, and hence gives very good estimation and detection rates for pitch estimation and voicing decisions.

In this work, we chose NCCF coefficients as the observation features O_t^q in an attempt to compare with "get_f0." With slight modifications to Equation (4), dozens of other score-based PDAs can be used as observation features as long as the maximum score is normalized to one. Similarly, observation features O_t^d are not confined to relative power change; spectrum change can also be integrated to help make local voicing decisions. Also, better quantization and clustering of pitch transition patterns can be further explored. With a more effective PDA or more robust pitch transition features, we believe performance will further improve over what has been reported in Section 3.

Although our approach gives efficient and robust estimation of model parameters, the decoding time could be a hindrance to its practical use as N scales up. This is because all N candidates of Q_t are involved in the inference of Equation (7) (also the DP of Equation (8)), and typically $N > 100$. One simple solution to this problem is beam search, where only the paths with sufficiently high likelihoods are kept in inference. Alternatively, a fixed number of top candidates are selected at each frame based on local costs, and they are the only ones considered in the DP. While no beam pruning was employed in this work, and while running time was perfectly acceptable, both methods above will lead to significant speedups without loss of performance.

One weakness of this framework is the lack of availability of enough training data with reliable ground truth. With the increasing use of laryngographs in speech analysis, however, more data will become readily available to use. The authors would like to thank Manhung Siu for providing us the database from the Hong Kong University of Science and Technology.

5. References

[1] D.Talkin, *Speech coding and synthesis*. Elsevier Science B.V, 1995.

[2] W.Hess, *Pitch Determination of Speech Signals*. Springer-Verlag, 1983.

[3] B.S.Atal, "Automatic speaker recognition based on pitch contours," *Journal of the Acoustical Society of America*, vol. 52, no. 6, 1972.

[4] N.Kunieda, T.Shimamura, and J.Suzuki, "Robust method of measurement offrundamental frequency by ACOLS-autocorrelation of log spectrum," in *Proc. Intl. Conf. on Acoustic, Speech and Signal Processing*, 1996.

[5] D-J.Liu and C-T.Lin, "Fundamental frequency estimation based on the joint time-frequency analysis of harmonic spectral structure," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 6, 2001.

[6] A.Cheveigne and H.Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, 2002.

[7] J.Droppo and A.Acero, "Maximum a posteriori pitch tracking," in *Proc. Intl. Conf. on Acoustic, Speech and Signal Processing*, 1998.

[8] J.Bilmes and G.Zweig, "The Graphical Models Toolkit: An open source software system for speech and time-series processing," in *Proc. Intl. Conf. on Acoustic, Speech and Signal Processing*, 2002.

[9] I.J.Good, *The estimation of probabilities: an essay on modern Bayesian methods*. MIT Press, Cambridge, MA, 1965.

[10] A.Wrench, "A multichannel/multispeaker articulatory database for continuous speech recognition research," in *Workshop on Phonetics and Phonology in ASR*, 2000.

[11] L.R.Rabiner, M.J.Cheng, and A.E.Rosenberg, "A comparative performance study of several pitch detection algorithms," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 24, 1976.