

Using Data Mining to Learn the Patterns of Pitch Variation in Chinese Speech

Tingshao Zhu & Wen Gao

*Institute of Computing Technology, Academia Sinica
Beijing, 100080. P.R. China*

Abstract Pitch model is very important in speech synthesis, and it mainly describes the variation of pitch. In order to synthesize speech with high intelligibility and naturalness, a system should be with an appropriate pitch model. We try to find the pitch model from actual speech samples by data mining. A prototype system called SpeechDM has been implemented to extract patterns from two-word phrase of Chinese. Dataset is used for data management, and multi-thread training tasks are controlled by training manager. This paper gives the process firstly, then each step is introduced in detail. Some results and conclusions are given at last.

1 Introduction

In the field of speech signal process, pitch is one of the most important parameters. Tone is decided by pitch variation, and it denotes different meaning in some languages especially in Chinese^[1]. When two or more Chinese syllables combine together, each syllable's tone will be changed. In speech synthesis, it is very useful to know the pattern of pitch variation to improve the intelligibility of synthesis speech^[2]. Although there are many pitch models, they cannot describe the variation completely, because they are made by hand through some statistics on some speech samples. These speech samples cannot cover all the speech phenomena. On the other hand, it is difficult to deal with massive speech data.

Since the pitch variation is very important for speech synthesis, the reasonable patterns of pitch variation can improve the quality of synthesis speech. It is naturally to think that maybe the patterns which are extracted

from actual speech samples can be helpful for speech synthesis. Because the size of speech database is always very large, it is impossible for people to extract these patterns by hand.

Knowledge Discovery in Database(KDD) is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understanding patterns in data^[3]. It employs statistical and computational technology to extract useful patterns from large database^[4]. KDD can also be called data mining, and the term data mining will be used in this paper.

In order to extract pitch variation patterns, a data mining system called SpeechDM has been implemented to extract these patterns. In the prototype, dataset is used to manage training and testing examples efficiently, and for scheduling the learning tasks, training manager is designed to control all of them.

2 SpeechDM Process

The prototype is used to deal with speech data and find the patterns of pitch variation. It consists of data preprocessing, data management, data mining and training management, and it provides tools to carry out these operations. Figure 1 depicts the process of SpeechDM.

SpeechDM separates end user from data miner by using of new data management method. For end users, they can extract and process data according to knowledge discovery tasks, and pay no attention to the learning algorithm. To data miner, they can concentrate on the development of new algorithm or the improvement of existed ones. The algorithm accesses training examples through the uniform interfaces which are predefined by the system. End users fully understand the requirements of task, and they can give the data scope exactly. For the data miner, they can test their algorithms to see which one is the best and thus can extract more efficient knowledge.

There are some steps in the whole process. Task analysis is done by data miner and domain experts to specify the data and learning target relate to the task. Algorithm selection aims at the optimize selection of algorithm according to the task. Data extraction accesses database for preparing the learning data. Data preprocessing manipulate the extracted data to fit for the requirement of learning. Dataset management administrates the dataset used by learning algorithms. Data mining extracts knowledge from data. Result analysis evaluate the result to see whether the task has been achieved. Outputting just displays the knowledge conveniently.

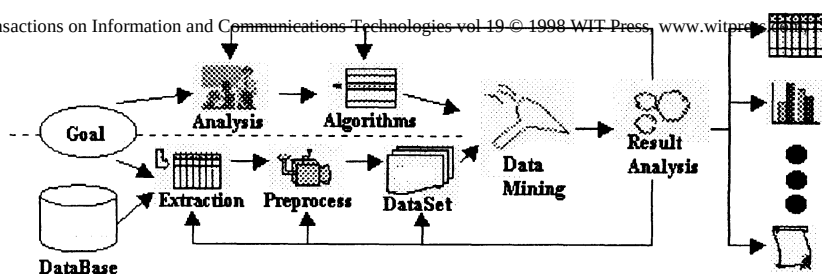


Figure 1: SpeechDM Process.

3 System Description

3.1 Data Preprocessing

The Speech Database that we are using is a Chinese speech synthesis database called CoSS-1. CoSS-1 includes the pronunciation of all isolate syllables, the 2-4 word phrases and some sentences. The number of isolate syllables with tone is 1268, and that of two-word phrase is 640. CoSS-1 records the speech wave and laryngograph synchronously. The sampling rate is 16000/s, and each sample is stored in two bytes. The data now being used are made by a young man. The two-word phrase covers almost the whole tone collocations in Chinese pronunciation.

To learn the patterns, the pitch should be calculated at first. There are many methods to extract pitch from speech wave, but the precision is very low^[5]. Since we want to learn the patterns and use them to generate pitch after training, the accuracy is very important.

A tool called Pitcher is implemented to extract pitch from laryngograph. It works by annotating each cycle's beginning and ending point, then calculating the pitch. let X_i be the beginning point of one cycle and X_j be the ending point, then the pitch of this cycle should be $16000/(X_j - X_i)$. Pitcher can also be used to split phrases and play the speech data.

In SpeechDM, Neural Network is used to learn the patterns. It is well known that the number of neural net's input units is fixed, but the length of pitch which acts as the training examples differs from each other significantly. A new algorithm is designed to wrap the pitches to make them suitable for the training.

For the speech data we used, the pitches' value domain is between 50 – 260. The following equation is used to normalize the pitch value.

$$\text{Normalized_Pitch} = (\text{Pitch_value} - \text{min}) / (\text{max} - \text{min})$$

Where **max** is the maximum of all pitches' value and **min** is the minimum. **Pitch_value** stores the pitch to be calculated and **Normalized_Pitch** is the normalized value.

3.2 Dataset Management

In SpeechDM we propose dataset to separate the data from learning algorithms, that is to say, make the data stand-alone. One dataset isn't used by only one special algorithm, and an algorithm can use many datasets according to data interface.

Dataset describes the data that relates to the learning task. It just gives the definition of the data referenced by knowledge extracting task, and has no relation with algorithm, and it also specifies the methods for the creation of training and testing examples. When one algorithm is used to extract knowledge, the training and testing data should be created firstly according to the methods in the description of the dataset. Since it only gives the scope of related data, there may be some redundant data, so the algorithm should make some decision for optimization.

To implement dataset, a structure is defined to describe the properties and the method to create the training and testing data. In SpeechDM, two data format have been defined, and each algorithm can access the training examples according to one of them. The dataset structure is following.

```
Struct DataSet{
    Char *name;           // Dataset name
    Char *comments;       // the comments of dataset
    Char *base;           // the used dataset base
    ...                   // other properties of dataset
    Int CreatNethod;      // method create training and testing data
}
```

A dataset manager is implemented for creating, deleting and example creation. To manage efficiently, we use dataset base to define the original scope, and it stores the data in one or more tables which can be

used by some operations. When the training and testing data are to be created, some operation can be executed to process the data from the base, such as filter and maintenance. The training and testing data then can be created from the processed dataset base.

3.3 Data Mining [7,8,9]

There are many kinds of neural networks which can be used for learning. We intend to learn the patterns from the mapping between the pitches in isolate syllable and those in phrase, since backpropagation network has implicit input layer and output layer, and it can also give very good result^[6], thus it is chosen to be trained in SpeechDM.

To learning the pitch variation patterns from the mapping, the pitches of the syllable in phrase are extracted firstly, and the pitches of the same syllable but in isolation are extracted too. The pitches from phrase are acted as output and those from isolation as input. When syllables are combined, their pitches are modified according to some tone variation rules, so their tones can be regarded as the factors that influence the pitches of the two syllables. The tones of the two syllables are also included in input layer.

Two networks are built to extract the patterns from all the two-word phrases in CoSS-1. One is designed to learn the patterns from all the first syllables in two-word phrases in CoSS-1, and the other is for the second ones. For the network learning the first syllables, its input layer consists of 67 units, and the hidden layer consists of 100 units. There are 36 units in output layer. The other network's input layer consists of 67 units, and the hidden layer consists of 100 units. For the output layer, it includes 42 units.

In order to generate training and testing data, you should split the phrase firstly, calculate the pitches, wrap the pitches to the same length and normalize pitches' value. Then the data can be used to train and test neural network. Most of them act as training examples and others as testing.

3.4 Training Manager

In our system, each algorithm is defined as a class including some properties and functions. Training task is defined as a learning thread class, and it is composed by the specification of algorithm and some interfaces for the system to schedule. In the thread class, the interface to all learning algorithm is also defined. The properties and function of the

thread class is following:

```
class TtrainAlgorithm {
    char algoname[20];           // algorithm name
    char comments[20];          // comments of the task
    int threadid;                // thread ID
    int CurrentTrainTime;        // display the progress of training
    char datasetname[20];        // dataset name
    union Algorithm {
        TBPTTrainPitch * pitchtrainBP;
        TBPTTrainDuration * durationtrainBP;
        ...
    } TrainAlgoList;            // all usable learning algorithm
    void Information();           // parameter setting of each algorithm
    void Initialize();           // initialize funtion of each algorithm
    void Execute();              // the mainfunction of thread
};
```

Based on the thread class, we design a training manager to control all learning tasks. The training manager acts as master thread to control the other learning algorithm. It can create the instance of training thread, and schedule the training task, such as pause, resume and stop. The progress of training task can also be displayed in the manager.

A new task should be created by the following steps: creating a new training thread object, this can be done by instantiating the training thread; choosing the suitable training algorithm, and set some parameters of the algorithm, such as thread ID, the algorithm name and dataset name; calling the initialize function of the selected algorithm for initialization; After the parameters have been set, the training manager will put the new thread into its thread array and display the progress of training when the task has been selected by user.

3.5 PostProcessing

A tool is designed to show the test examples graphically. From the graph, it is easy to see whether the calculated pitch coincides with the actual pitch or not, and it can display as much as ten syllables' pitch at the same time. Figure 2 depicts one of the testing results.

For iterative learning, the dataset and train manager can be used to

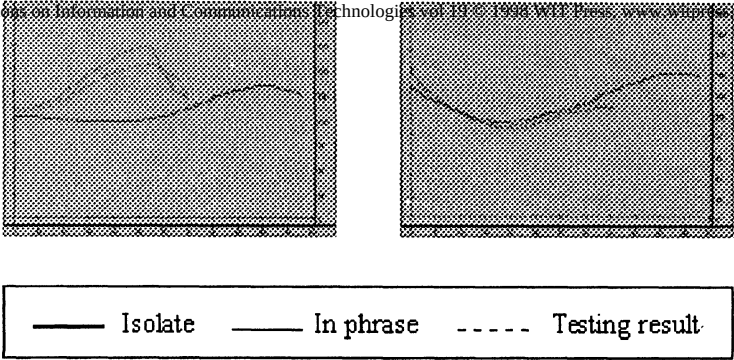


Figure 2: Result of gu3ban3.

refine the data and patterns learning. In our experiment, We found that when we take all the first syllables and the second ones for training a net, the result is poor. So two new datasets are created separately, and after training the new result is better than the previous. In refining, only new datasets are defined, the training program can be used without modification.

4 Result

Table 1 gives some statistics of the results from one test. The calculations are based on the deduction between the original syllable's pitches in phrase and those calculated by the network, and of course they have been wrapped to the same length.

	Phrase	Max	Min	Mean	Variance
F I R S T	Chuang1shan4	4	0	-1.477206	4.376950
	Jin4zhan4	13	0	-0.241210	42.273769
	Liu2xie4	18	0	-4.525616	15.284344
	Gong1hui4	15	0	2.656483	10.662880
	Wang2pai2	9	0	2.016155	7.791716
S E C O N D	Zhi4xun2	6	0	-0.001093	2.958961
	Bing3yao4	14	0	-3.638539	23.004646
	Bao3xian3	7	0	-0.527984	10.082108
	La1suo3	7	0	0.168661	7.902920
	Xiao4you3	19	0	4.453946	24.817568

Table 1: The results of one test.

5 Conclusion

SpeechDM is the data mining system which is designed to learn the patterns of pitch variation in Chinese two-word phrase. It provides tools which can be used in the stage of preprocessing, data management, data mining and training management. The dataset and multi-thread training technologies have made the system easy to extract knowledge and extend. We hope that by learning from actual speech examples, it is possible to improve the intelligibility and naturalness of Chinese synthesis speech.

REFERENCES

1. Lin Tao, Wang Lijia, Acoustics course of study, Peking University Press, 1994.
2. Chu Min, Research on Chinese TTS system with high intelligibility and naturalness, Ph.D thesis, Institute of Acoustics, Academia Sinica, September, 1995.
3. Usama M.Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasamy Uthurusamy, EDITORS, Advance In Knowledge Discovery And Data Mining, AAAI/MIT Press, 1996.
4. George H.John, Enhancements to the Data Mining Process, Ph.D thesis of Stanford University, 1997.
5. Yang Xingjun, Chi Huisheng, Speech Signal Digital Process, Publishing House of Electronic Industry, 1990.
6. Wang Wei, Principle of Artificial Neural Network ---- rudiment and implement, Beijing University of Aeronautics and Astronautics Press, 1995.
7. Kero, B, L. Russell, S. Tsur, W.M. Shen, An Overview of Data Mining Technologies, The KDD Workshop in the 4th International Conference on Deductive and Object-Oriented Databases, Singapore, 1995.
8. Famili, The Role of Data Pre-processing in Intelligent Data Analysis, Proceedings of the IDA-95 Symposium, Baden-Baden, Germany. P. 54-58, 1995.
9. J. Han, Y. Fu, Y. Huang, Y. Cai, N. Cercone, DBLearn: A system prototype for knowledge discovery in relational databases, Proc. 1994 ACM-SIGMOD Int'l Conf. on Management of Data (SIGMOD'94), Minneapolis, MN, May, 1994.