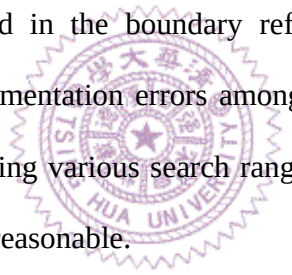
Chapter 6. Boundary Refinement Based on a Score Predictive Model

So far, we have demonstrated the performance of the proposed hybrid approach which effectively combines statistics-based and heuristic methods together to refine the boundaries. Although the overall segmentation accuracy is increased via the hybrid method, there is still room for further improvement. In fact, the proposed hybrid approach has two weaknesses:

1) It is somewhat unnatural to have a binary decision for crisp classification. A soft or fuzzy classification might be more desirable since the degree of a boundary belonging to a certain class is represented by a continuous number between 0 and 1.
2) A fixed search range used in the boundary refinement is inappropriate. This is because that the initial segmentation errors among all phonetic transition categories are usually diverse, i.e., using various search ranges for different phonetic transition categories should be more reasonable.



According to the two viewpoints mentioned above, we shall address the use of soft classification based on the concept of the score predictive model (SPM). The principal advantage of the proposed SPM is its capability to predict the scores of candidate boundaries reliably. Under the framework of the SPM we need to set up a reasonable score function. For each phonetic transition category a set of candidate boundaries are collected in advance. These boundaries are transformed into a training set that contains acoustic features and desired scores. Here the acoustic features are the same 58-dimensional feature vectors used in the previous SKL boundary refinement. Finally, a reliable regression approach is employed to construct the SPM based on supervised learning.

In the process of boundary refinement, the scores of two initial boundaries identified by DTW and the HMM are computed via the SPM. The boundary with the higher score is preserved and the other is discarded. A dynamic search range is designed to determine the suitable candidate boundaries around the preserved boundary. Subsequently, the score of each candidate boundary is computed through the SPM. Finally, a boundary with the highest score will be selected as the refined boundary. The following subsections shall introduce the SPM in detail.

6.1. Score Function

The score function is used to specify the score of a boundary according to its distance from a true boundary. Intuitively, if a candidate boundary is closer to the true boundary it should receive a higher score. Based on this concept, the score function for the k^{th} phonetic transition category is defined by a Gaussian-like function:

$$score(d, \sigma_k) = 100 e^{-\frac{Cd^2}{\sigma_k^2}}, \qquad (6.1)$$

where d denotes the distance in ms between a true boundary and a candidate boundary. The value σ_k is set to the maximum segmentation error (in ms) of the k^{th} phonetic transition category. However, we also set the upper bound of σ_k to four times the standard deviation of segmentation error in order to reduce the influence of possible outliers. The constant coefficient C is specified as 20 to set the score for the maximum segmentation error to approximate zero. The use of the continuous score function is intuitive and more reasonable than the crisp classification used in previous approaches since the concept is similar to a soft

or fuzzy classification. Fig. 1 shows the score function of the first phonetic transition category which is composed of the transition from the first FINAL type to the first INITIAL type with a maximum segmentation error of 227.9375 ms. So, if a candidate boundary is 28 ms away from the true boundary, then the corresponding score is $100e^{-\frac{20(28)^2}{(227.9375)^2}} = 73.9488$.

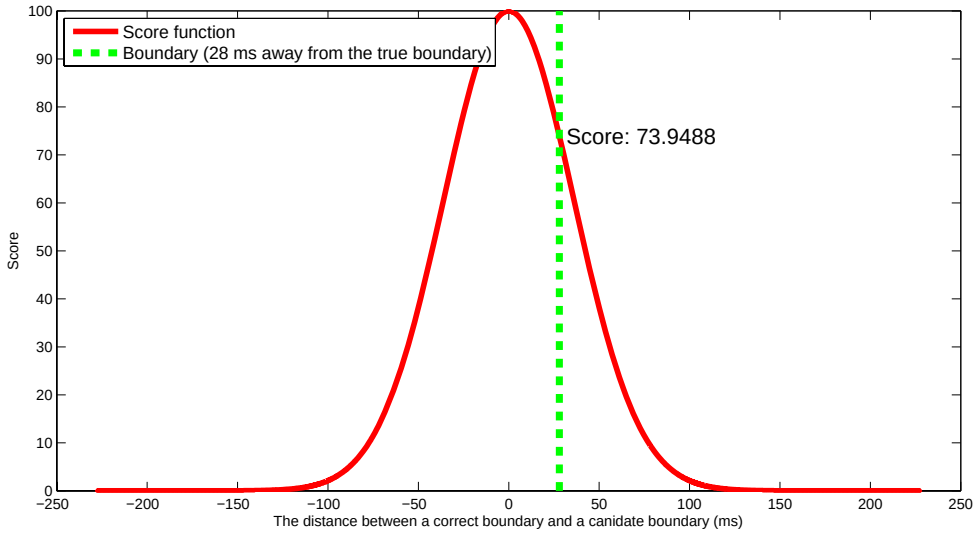


Fig. 6.1. The score function of the first phonetic transition category.

6.2. Candidate Boundaries for Training

Once the score function is defined, a score predictive model (SPM) is subsequently constructed for each phonetic transition category. As documented in Chapter 5, the data of TTS-455 within ±50 ms segmentation error tolerance is approximately 96 percent while applying the HMM-based initial phonetic segmentation. As a result, given a true boundary, the candidate boundaries located within ±50 ms of this boundary are collected to form the training data. These candidate boundaries are collected through the following procedure:

1) Add a set of candidate boundaries 2 ms apart, located within ±10 ms of the true boundary.
2) Add a set of candidate boundaries 5 ms apart, located within 10 ~ 50 ms and -10 ~ -50 ms around the true boundary.

As a result, a total of 27 candidate boundaries (including the true boundary) can be collected for each true boundary, as shown in Fig. 6.2. This task of training data collection is repeated in order to obtain 54 sets of training data corresponding to the 54 possible phonetic transition categories.

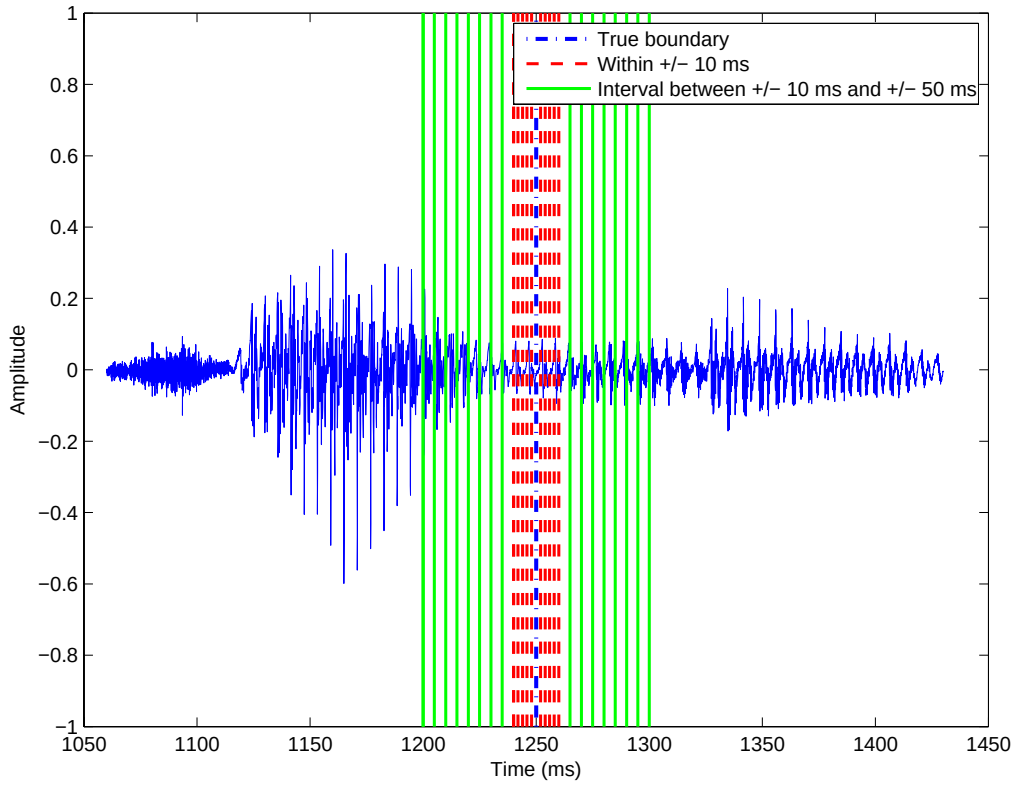


Fig. 6.2. A typical example of the 27 candidate boundaries around a human-labeled true boundary. The content of this speech waveform was “將離” (“jiang-li2”).

On the other hand, the accuracy of SVS-1384 within ±200 ms segmentation error tolerance is approximately 97 percent using both the HMM-based and DTW-based initial phonetic segmentation. Hence the candidate boundaries located within ±200 ms of the true boundary are collected to form the training data through the following procedure:

1) Add a set of candidate boundaries 4 ms apart, located within ±40 ms of the true boundary.
2) Add a set of candidate boundaries 16 ms apart, located within 40 ~ 200 ms and -40 ~ -200 ms around the true boundary.

As a result, a total of 41 candidate boundaries (including the true boundary) can be collected for each true boundary, as shown in Fig. 6.3. This task of training data collection is repeated in order to obtain 54 sets of training data corresponding to the 54 possible phonetic transition categories.

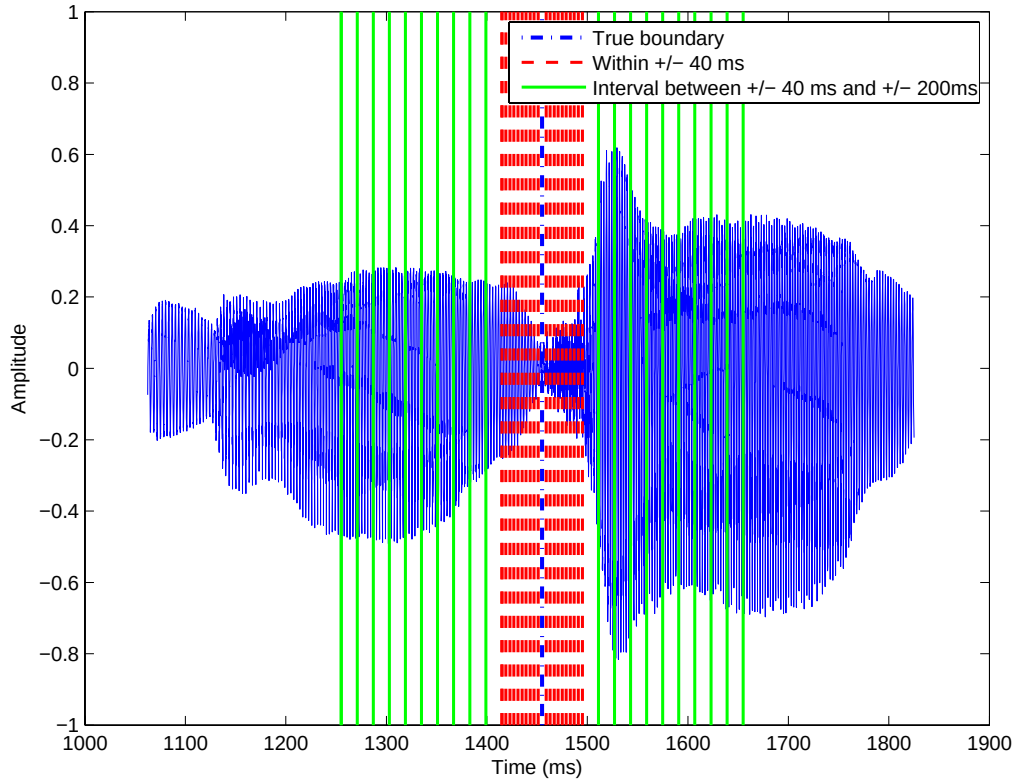


Fig. 6.3. A typical example of the 41 candidate boundaries around a human-labeled true boundary. The content of this singing voice waveform was "寧靜" ("ning2-jing4").

6.3. Regression Model by Using Support Vector Machine

After collecting the training data, we need to adopt a regression approach to construct a SPM for each phonetic transition category. Generally speaking there are many approaches for regression, such as linear regression (LR), neural network (NN) [64], support vector machine (SVM) [65], etc. In this study, we chose SVM to construct SPMs by using the library provided by LIBSVM [67]. The following is a brief description of how SVM works on regression problems.

The SVM algorithm is based on the statistical learning theory. For regression problems, the principal goal of the SVM is to construct a hyperplane that is close to as many of the data

points as possible. The two commonly used SVM methods for regression problems are the ε-support vector regression (ε-SVR) [68] and the ν-support vector regression (ν-SVR) [69]. Given a set of data points, $\{(\mathbf{x}_i, y_i)\}_{i=1}^{k}$, where $\mathbf{x}_i \in \mathrm{R}^n$ is an input vector and $y_i \in \mathrm{R}^1$ is a target output value, our objective is to find the coefficients of a hyperplane which can minimize an objective function consisting of the sum of the squared norm of the hyperplane's coefficient and the total distances from the data points to the hyperplane using Vapnik's ε-insensitive loss function. The ε-insensitive loss function is defined as:

$$\left|y_i - (\mathbf{w}\mathbf{x}_i + b)\right|_\varepsilon = \begin{cases} 0 & \text{if } \left|y_i - (\mathbf{w}\mathbf{x}_i + b)\right| \le \varepsilon \\ \left|y_i - (\mathbf{w}\mathbf{x}_i + b)\right| - \varepsilon & \text{otherwise} \end{cases}. \tag{6.2}$$

The parameter ε is determined by the user. The primal problem of ε-SVR is as follows:

$$\underset{\mathbf{w},\xi_i,,\xi_i^*}{\text{minimize}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\left(\frac{1}{k}\sum_{i=1}^{k}\left(\xi_i + \xi_i^*\right)\right), \tag{6.3}$$

$$\text{subject to}: \begin{cases} \left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) - y_i \le \varepsilon + \xi_i, \\ y_i - \left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0, i = 1,...,k. \end{cases} \tag{6.4}$$

Here training vectors $\mathbf{x}_i$ are mapped into a higher dimensional space through a kernel function ϕ. The inequality constraints embed Vapnik's ε-insensitive loss function, which indicates that if $\mathbf{w}^T\phi(\mathbf{x})$ is in the range of $[y-\varepsilon, y+\varepsilon]$ no loss is considered. Two slack variables ξ_i and ξ_i^* are introduced, one for exceeding the target value by more than ε, and the other for being more than ε below the target. In addition, *C* denotes the penalty parameter; a larger *C* corresponds to a higher penalty being assigned to training errors. Fig.

6.4 shows an example of ε-SVR.

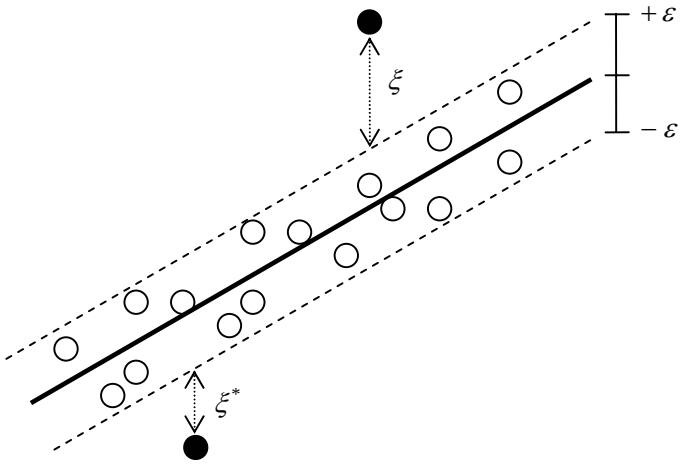


Fig. 6.4. The regression function of ε-SVR is represented by a tube with radius ε and slack variables ξ_i. The data points outside the ε-insensitive zone are referred to as support vectors (black dots).

Since it is difficult to find an appropriate value for ε, Schölkopf et al. [69] proposed ν-support vector regression using a parameter ν which effectively controls the number of support vectors and training errors. The primal problem of ν-SVR is as follows:

$$\underset{\mathbf{w},\varepsilon,\xi_i,\xi_i^*}{\text{minimize}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\left(\nu\varepsilon + \frac{1}{k}\sum_{i=1}^{k}\left(\xi_i + \xi_i^*\right)\right), \qquad (6.5)$$

subject to (6.4) and $\varepsilon \geq 0$. They proved that $\nu \in (0,1]$ is an upper bound of the fraction of margin errors and a lower bound of the fraction of support vectors. In our study, ν was empirically set to 0.5 where it gave a satisfactory performance based on our observations. In the present study we employed ν-support vector regression and adopted the radial basis function (RBF) as its kernel function whose equation is shown as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2} \quad (6.6)$$

.

In (6.5) and (6.6), there are two significant parameters, *C*, and *γ,* to be determined to construct a good regression model. A common approach is to adopt a grid search with n-fold cross validation. Here we adopt a two-phase grid search with five-fold cross validation to find the best (*C*, *γ*) pairs for the SPM of each phonetic transition category. The two-phase grid search is described as follows.

1) Phase 1: A coarse grid search with five-fold cross validation is applied to find a reference point for the next finer grid search. That is, a set of parameters ($C = \{2^1, 2^3, ..., 2^{11}\}, \gamma = \{2^{-13}, 2^{-11}, ..., 2^{-1}\}$) are evaluated to find the best (*C*, *γ*) as the reference point that has the smallest root mean square error.
2) Phase 2: In the neighborhood of the reference point identified in phase 1 we conduct a finer grid search with five-fold cross validation to find a refined point (*C*, *γ*).

Fig. 6.5 shows the flowchart of the SPM construction for each of the 54 phonetic transition categories.

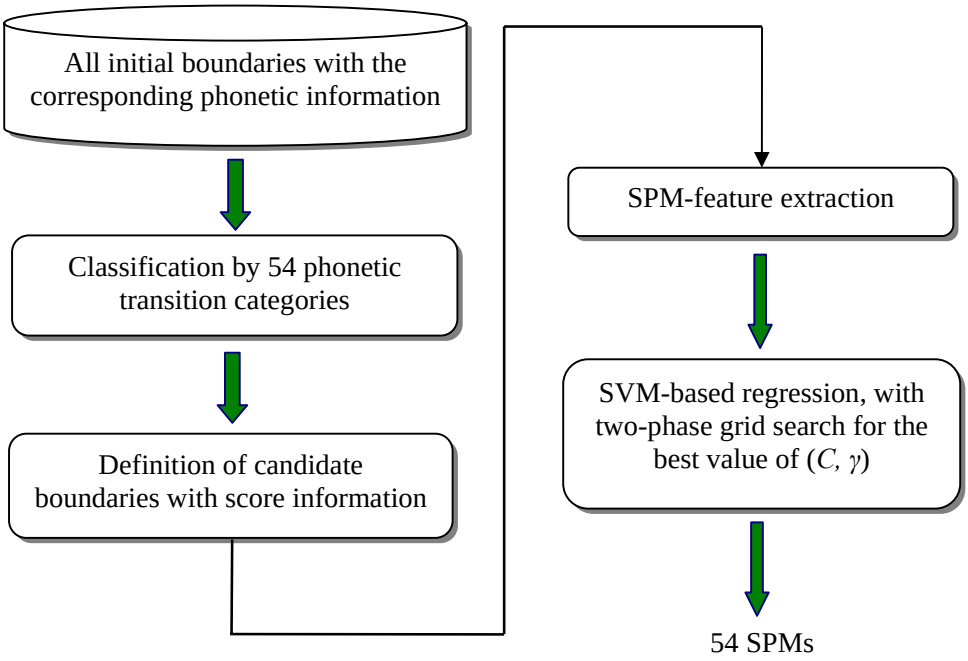


Fig. 6.5. The construction of 54 SPMs.

6.4. Boundary Refinement by Using SPM

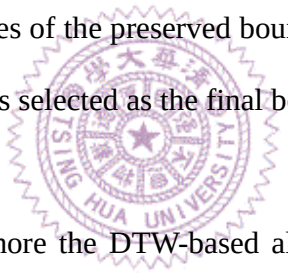
As mentioned previously, the proposed SPM can take advantage of the results of the HMM and the DTW to obtain better segmentation, as shown in Fig. 6.6. The overall segmentation procedure for singing voice corpora is summarized as follows.

1) For each phoneme boundary between two consecutive syllables, there are two initial estimates obtained from the HMM and the DTW.
2) The corresponding SPM is used to predict the scores of two initial estimates by HMM and DTW. The boundary with a higher score is preserved and the other is discarded.
3) A dynamic search range for refinement is determined according to the score of the preserved boundary. The size of the search range can be calculated via (6.7) which is derived from (6.1).

$$Search \quad range = \sqrt{\frac{\sigma_k}{20} \log\left(\frac{100}{score}\right)} \,. \qquad (6.7)$$

In other words, a lower score requires a wider search range, while a higher score requires a narrower one. It should be noted that a wider search range does not always guarantee a better performance since outliers could be introduced which then would produce unpredictable results via the SPM. Since the score is obtained by the regression model in the SPM, its value could possibly be out of the range of [0, 100]. To avoid this, we limit the range of the score to be within [1, 100] in order to find a reasonable search region in (6.7).

4) Finally, for the preserved boundary we select candidate boundaries 2 ms apart, within the search range at both sides of the preserved boundary. The candidate boundary that has the highest SPM score is selected as the final boundary.



It should be noted that we ignore the DTW-based alignment while the proposed SPM boundary refinement is performed on TTS-455 due to the fact that it has no corresponding music score information.

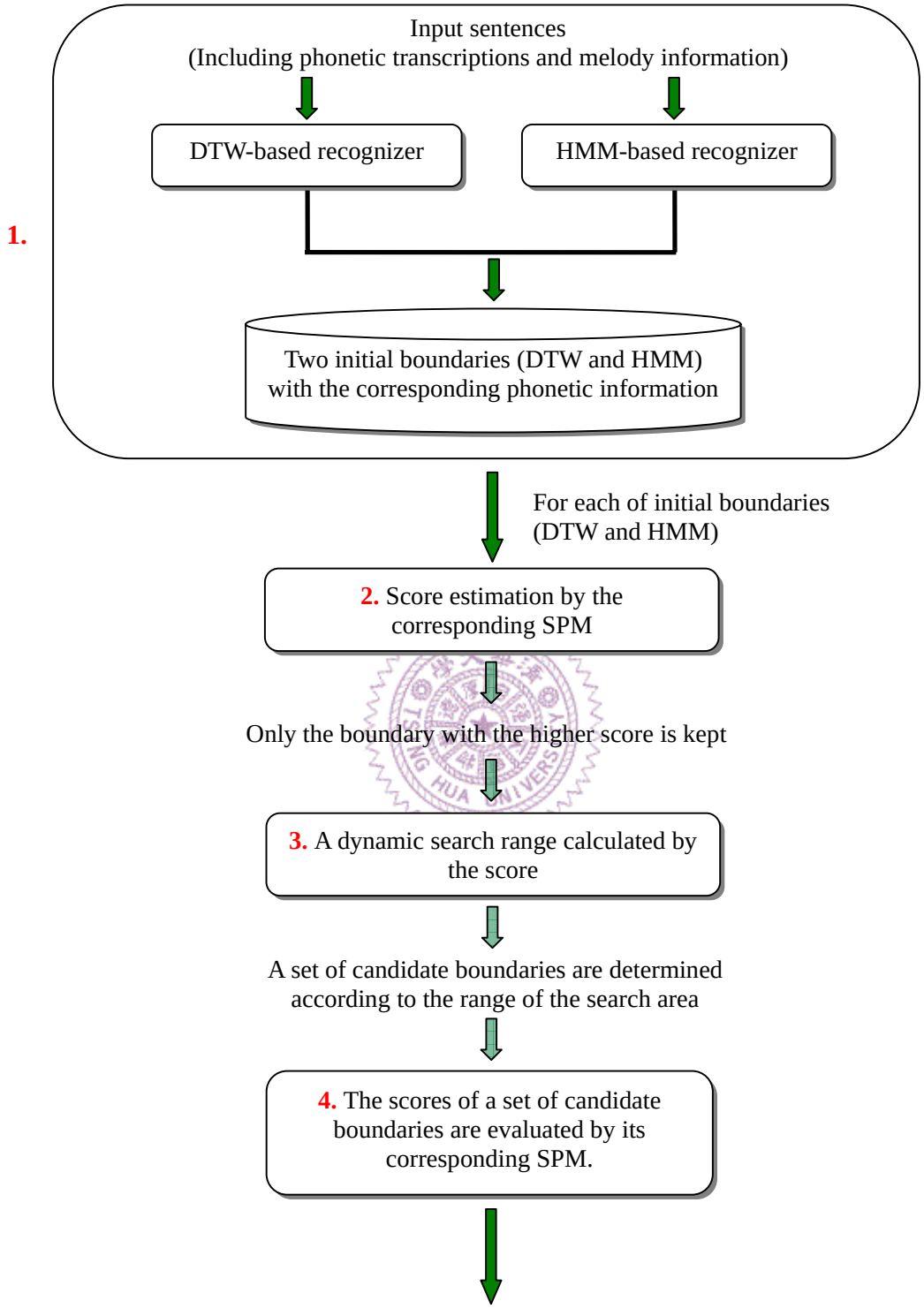


Fig. 6.6. Boundary refinement using the proposed SPM.

6.5. Performance Evaluation of SPM

In the section, we conducted several experiments to validate the feasibility of the proposed SPM. At the beginning, we applied SVM regression to construct the 54 SPMs. The best parameters (C, γ) for the 54 SPMs were determined via the two-phase grid search with five-fold cross validation. Fig. 6.7 demonstrates the results of the boundary refinement based on the proposed SPM, where the top and bottom panels plot the result of the closed and the open tests, respectively. It should be noted that the configurations of this experiment are the same with those of the previous experiment by using the hybrid approach, i.e., we used 300 sentences for the training set and 155 sentences for the test set for the performance evaluation of SPM on TTS-455. It can be seen from Fig. 6.7 that the results indicate that the proposed SPM is able to effectively improve the initial segmentation results obtained from the HMM.

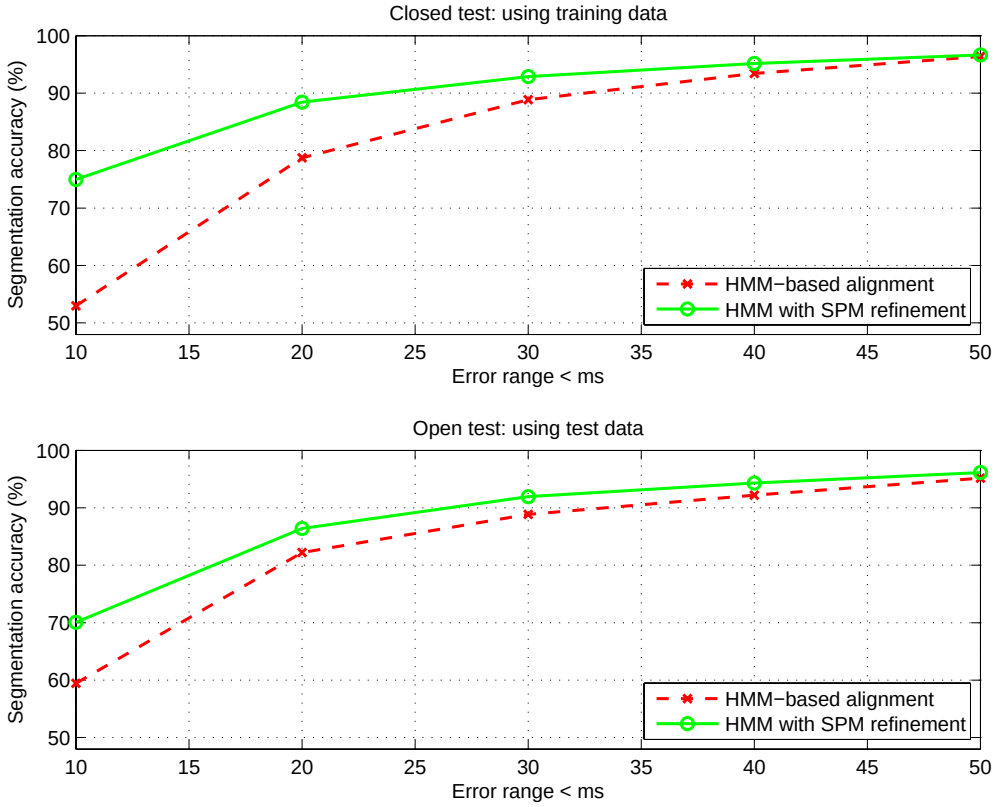


Fig. 6.7. The performance of the proposed SPM approach. Top: closed test. Bottom: open test. (Evaluted data: TTS-455)

On the other hand, we used the same procedure to evaluate the performance of the proposed SPM on SVS-1384. It can be seen from Fig. 6.8 that both the performance of DTW and the HMM are not satisfactory. In addition, the results indicate that the proposed SPM is able to effectively improve the initial segmentation results obtained from DTW and the HMM. For example, the percentage of the cases whose error range < 20 ms is increased from 67.2% (DTW-based alignment) to 76.6% (HMM+DTW with SPM refinement) in the open test.

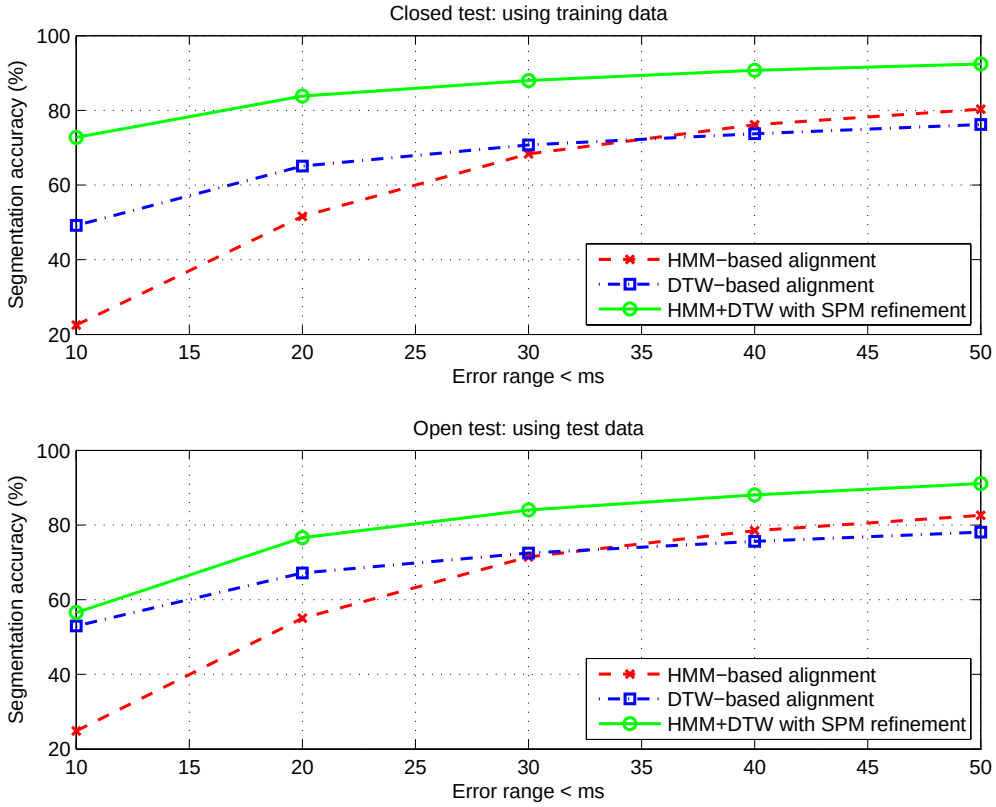


Fig. 6.8. The performance of the proposed SPM approach. Top: closed test. Bottom: open test. (Evaluted data: SVS-1384)

6.6. Performance Comparison Using Three Regression Approaches

In the previous experiments, although we have shown that SPMs constructed by SVM are able to obtain satisfactory results, it is still unknown whether other regression approaches will outperform SVM. In the study we tried two other common regression approaches, linear regression (LR) and neural network (NN), and compared them with SVM.

For LR we applied the least-squares error criterion [70] for obtaining the corresponding SPMs. The major advantage of LR is its efficiency in computing the least-squares solution.

The NN approach has a lot of parameters that must be determined, such as initialization of weights, number of neurons, learning rates, training methods, transfer functions, stopping

rules, number of training iterations, and so on. To find the optimum values of these parameters is quite time consuming. In this study, we chose the Levenberg-Marquardt method [71] for training the neural network of multilayer perceptrons (MLPs). The transfer function of the hidden layer was the hyperbolic tangent function. In the output layer the linear function was used. In theory, an NN with a single hidden layer can be proven to be a universal function approximator. However, more hidden layers may provide better performance through richer internal representation. In addition, the number of neurons also affects the overall performance. Therefore we chose several different configurations of NN, including a single hidden layer with 20, 40 and 60 neurons, and two hidden layers with 10, 20 and 30 neurons each. Subsequently, we introduced the five-fold cross validation to find the most suitable settings for NN. In addition, in order to avoid premature convergence to local minima, we trained each NN 10 times starting from different sets of random weights. After a lengthy training process, each SPM had its regression model of NN with different optimum structures and parameters.

In this dissertation, we conducted several experiments to compare the performance of different regression methods on two corpora, TTS-455 and SVS-1384. Fig. 6.9 demonstrates that both SVM and LR have better performance as compared with NN while refining the data of TTS-455. Fig. 6.10 shows that SVM has the optimal performance among the three regression methods while refining the data of SVS-1384. Consequently, we chose SVM instead of LR or NN to construct SPMs. Besides, there are some other reasons that make SVM a better choice than NN to construct SPMs, and they are:

1) An iterative procedure is used to reduce the possibility of premature convergence to local minima while training NN. However, the training problem of SVM is a convex optimization problem without local minima [72], and so there is no need to employ an additional iterative procedure for SVM.

2) SVM has the advantage that it is able to deal efficiently with high dimensional input vectors. On the contrary, the number of weights for a NN is very high in cases with high dimensional input vectors.

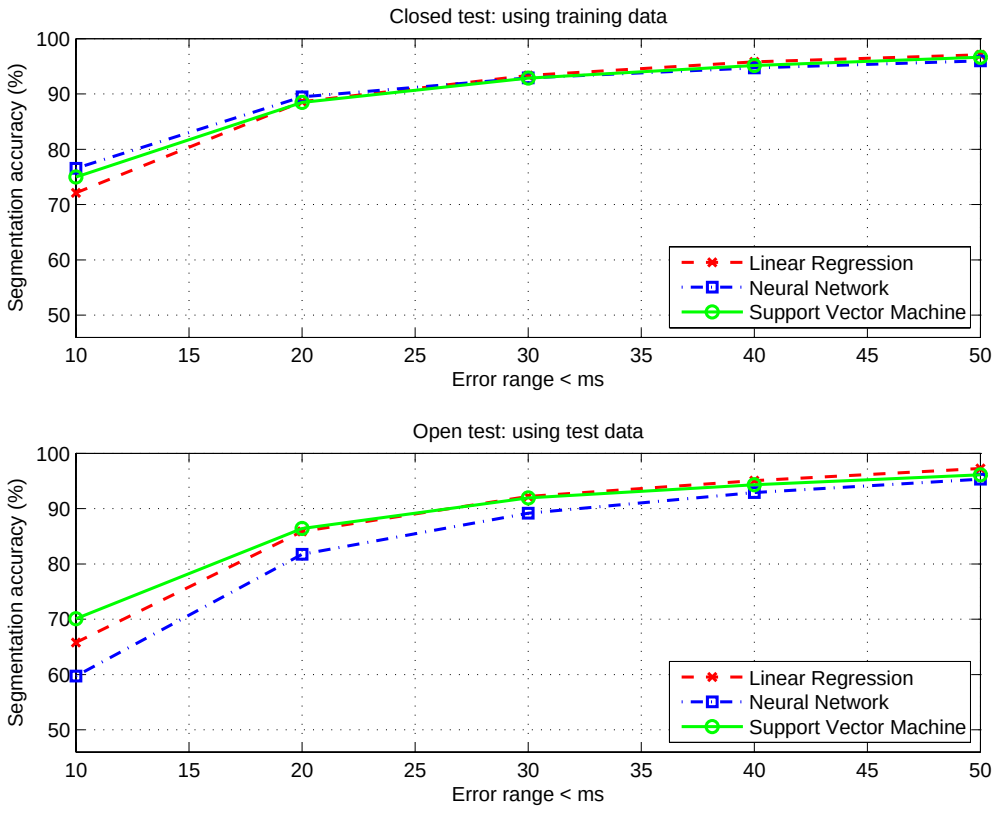


Fig. 6.9. Performance comparison using different regression approaches. Top: closed test. Bottom: open test. (Evaluted data: TTS-455)

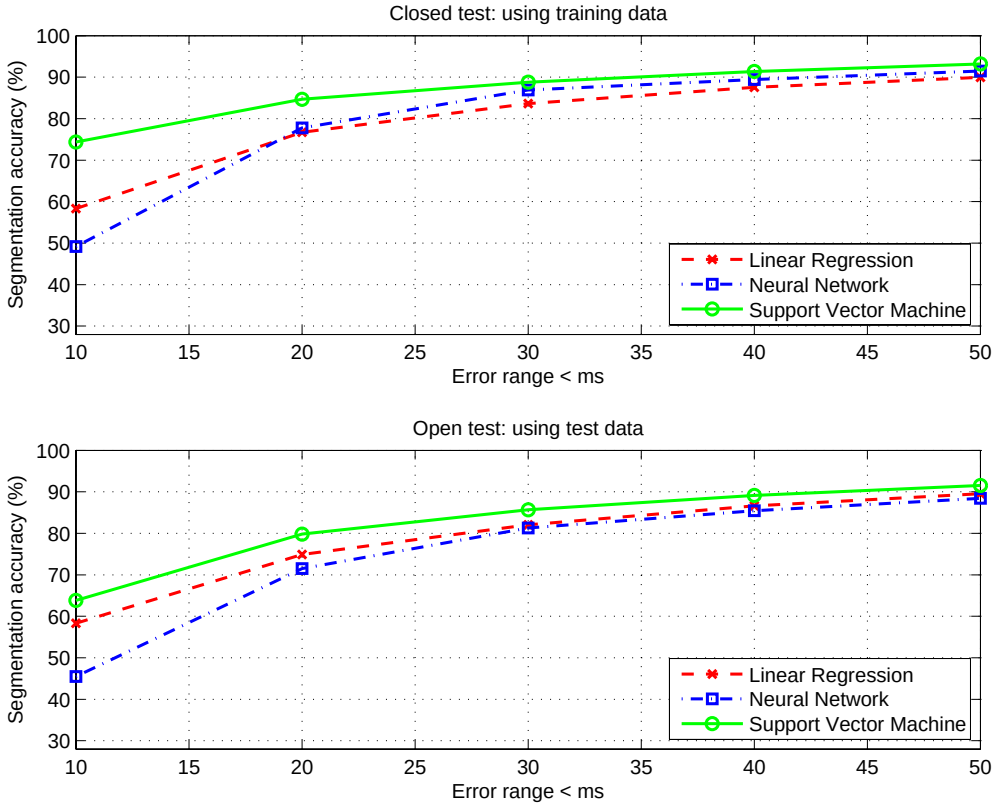


Fig. 6.10. Performance comparison using different regression approaches. Top: closed test. Bottom: open test. (Evaluted data: SVS-1384)

6.7. Performance Comparison Using Different Boundary Refinement Methods

In this section, a boundary refinement based on a hybrid approach was used for comparison with the proposed SPM. Fig. 6.11 and Fig. 6.12 present the performance comparison between the hybrid approach and the SPM approach. The evaluation data for the first experiment shown in Fig. 6.11 and the second experiment demonstrated in Fig. 6.12 are TTS-455 and SVS-1384, respectively. It can be seen from the open test in Fig. 6.11 that the performance of the proposed SPM approach is almost the same with that of the hybrid approach. However, the proposed SPM approach obviously outperforms the hybrid approach according to the experimental results demonstrated in Fig. 6.12. In other words, the

segmental results via the SPM approach are more reliable than those obtained by the previous hybrid approach.

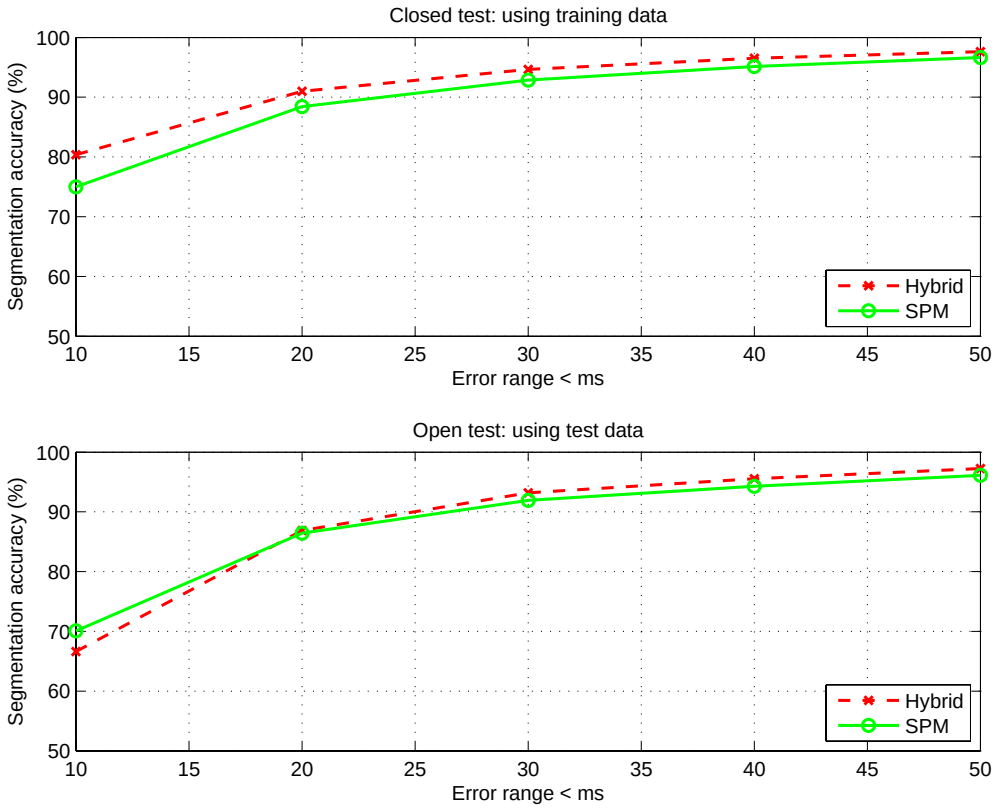


Fig. 6.11. Performance comparison between two boundary refinement approaches. Top: closed test. Bottom: open test. (Evaluted data: TTS-455)

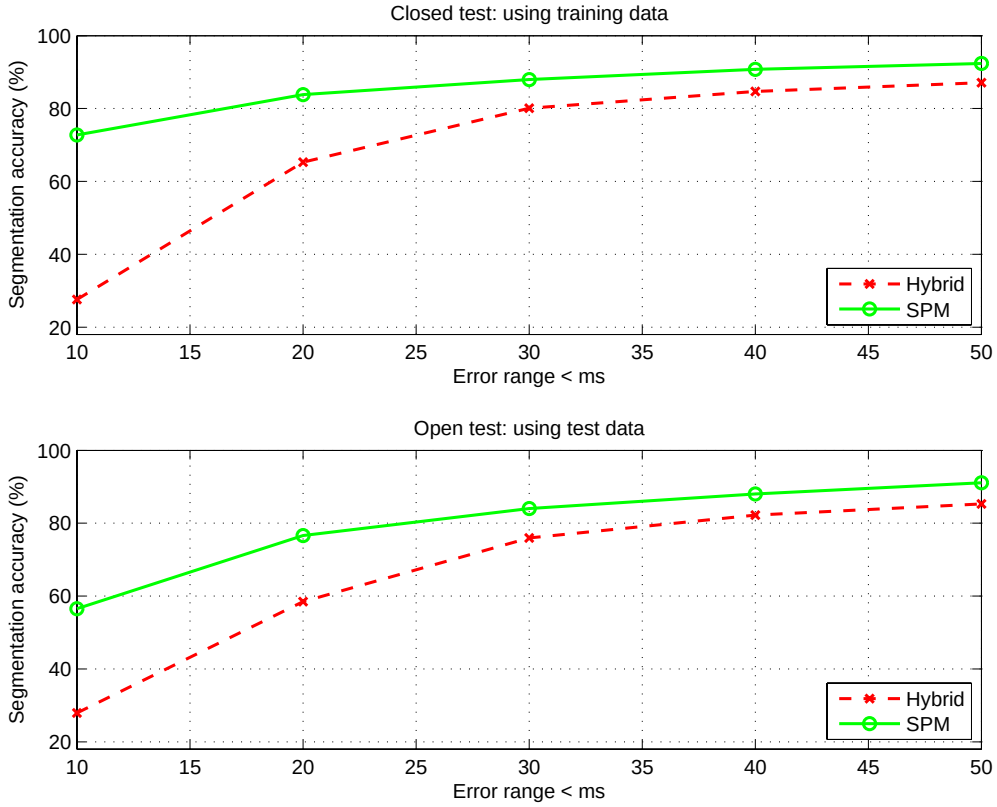


Fig. 6.12. Performance comparison between two boundary refinement approaches. Top: closed test. Bottom: open test. (Evaluted data: SVS-1384)

6.8. Two Attempts Regarding Performance Improvement

So far, we have conducted several experiments to demonstrate the feasibility of the proposed SPM. The experimental results indicate that the SPM indeed improves the segmentation accuracy. In fact, in this study we also tried two methods to observe whether the overall performance could be ameliorated further. The principal ideas of the two methods are described as follows.

1) Utilize more acoustic features instead of single feature (ex. MFCCs alone) to construct the HMM-based recognizer for subsequent initial phonetic segmentation.

2) Apply an additional procedure to effectively reduce the training errors while using the regression model to construct each SPM.

As for the above first method, we adopted the 58-dimensional feature vector used in the hybrid/SPM boundary refinement instead of 39-dimensional MFCCs to construct the HMM-based recognizer. For simplicity in the following performance comparison, we only construct a new speaker independent model by using TCC-300 corpus; other models in different types (ex. speaker dependent or speaker adapted) are not discussed in this dissertation. The newly trained model is used to perform the initial phonetic segmentation on both TTS-455 and SVS-1384.

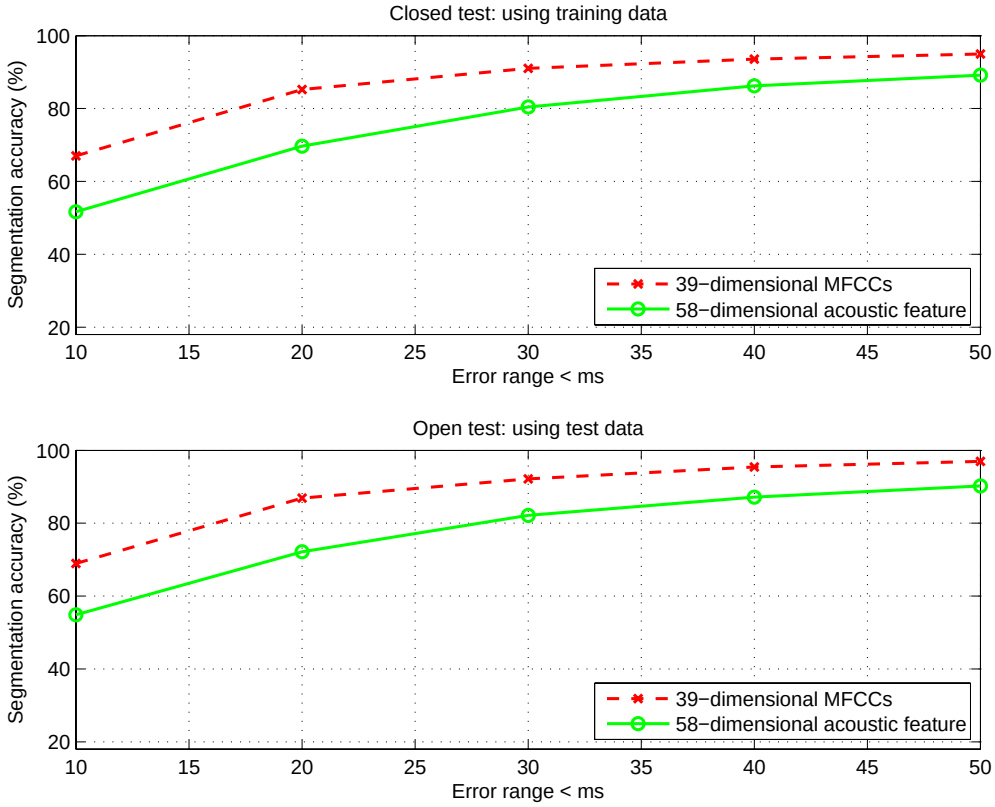


Fig. 6.13. Performance comparison between two kinds of acoustic features. Top: closed test. Bottom: open test. (Evaluted data: TTS-455)

From Fig. 6.13 and Fig. 6.14, the experimental results reveal that the new HMM-based recognizer reduces the segmentation accuracy greatly. This is probably caused by that the 58-dimensional feature is not suitable for speech recognition as compared with the traditional 39-dimensional MFCCs. For example, pitch is not an ideal feature for speech recognition since different syllables possibly have the same pitch. Based on most of experimental results, the low speech recognition rate usually accompanies low segmentation accuracy.

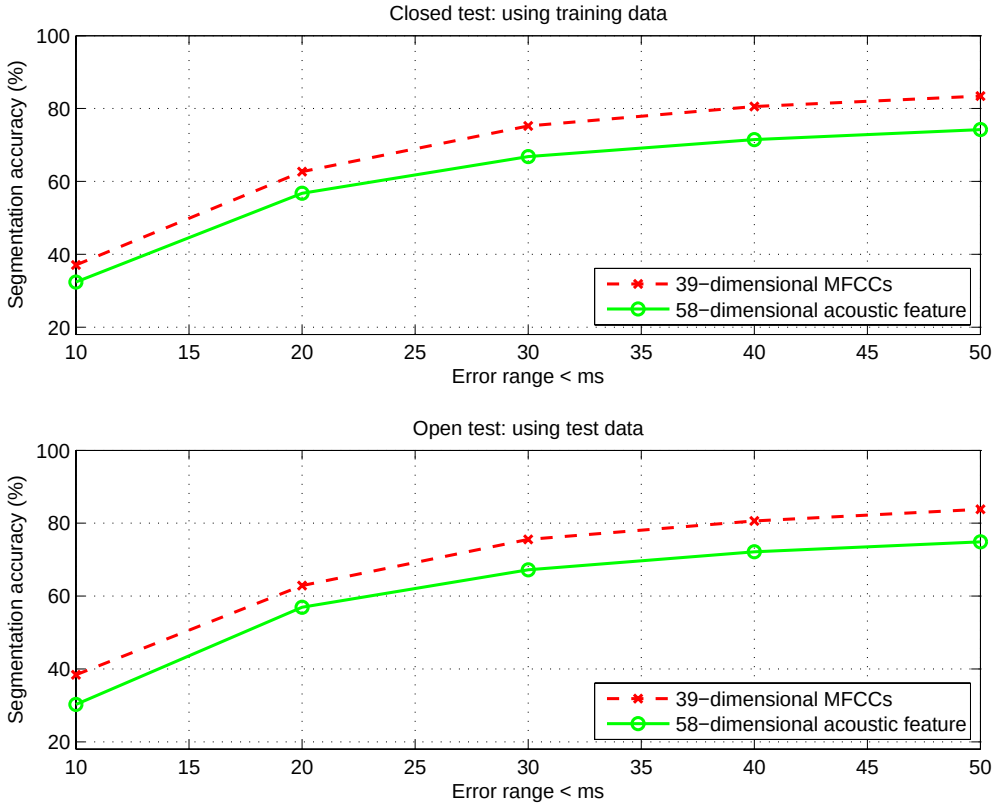


Fig. 6.14. Performance comparison between two kinds of acoustic features. Top: closed test. Bottom: open test. (Evaluted data: SVS-1384)

As for the second method, we employ Lee's method [29] to investigate the possibility of further performance improvement. Lee proposed a joint classification/training algorithm to find an optimum set of multilayer perceptrons (MLPs) for the optimum partition of phonetic transition categories. The algorithm involves the following steps.

Step-0. Initialization: Given a training set and initial partitions, construct the initial MLP for each partition. (A partition is a set of phoneme-phoneme combinations.)

Step-1. Classification: For the training data of each phoneme combination, find the index of the minimum-error MLP.

Step-2. Partitioning: The training data of each phoneme combination is partitioned according to the minimum-error MLP index obtained in *Step-1*.

Step-3. Retraining MLPs: Each MLP is retrained for each partition (with updated phoneme combination and corresponding training data) for better performance.

Step-4. Convergence test: Compute the overall error E_i at the *i*th retraining stage. If $(E_{i-1} - E_i)/E_{i-1} \le \varepsilon$, stop; otherwise, replace *i* by *i*+1 and go to *Step-1*.

In Lee's study, only four phonetic transition categories were used based on the voicing status of the phoneme [25]. Then the algorithm iteratively updated the optimum set of MLPs and the partitioning for the four phonetic transition categories. In the study we adopted the 54 phonetic transition categories mentioned previously and then iteratively updated the optimum set of SPMs for these categories by employing the joint classification/training algorithm. These SPMs were constructed by SVM instead of NN because of the advantages provided by SVM, which were addressed earlier in this dissertation.

The experimental results are shown in Fig. 6.15 and Fig. 6.16, in which the addition of Lee's algorithm gives slightly better performance for the closed test. However, the performance difference between SPM and SPM with Lee's method is almost the same. These results were expected since the original phonetic transition categories are already close to the optimum partitioning because they were manually selected based on some acoustic properties.

Although the two kinds of methods mentioned above fail in improving the overall performance effectively, they are probably useful if we combined other influential acoustic features (for the first method) or if the larger training data was available (for the second

method).

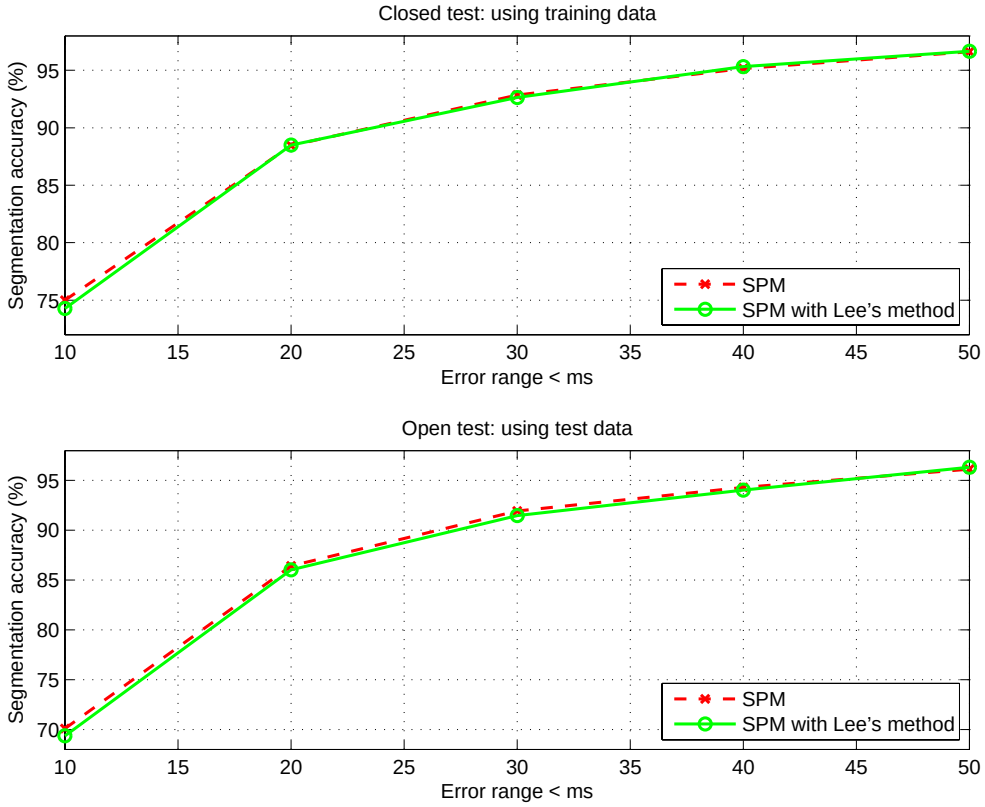


Fig. 6.15. Performance comparison of SPM and SPM combined with Lee's method. Top: closed test. Bottom: open test. (Evaluted data: TTS-455)

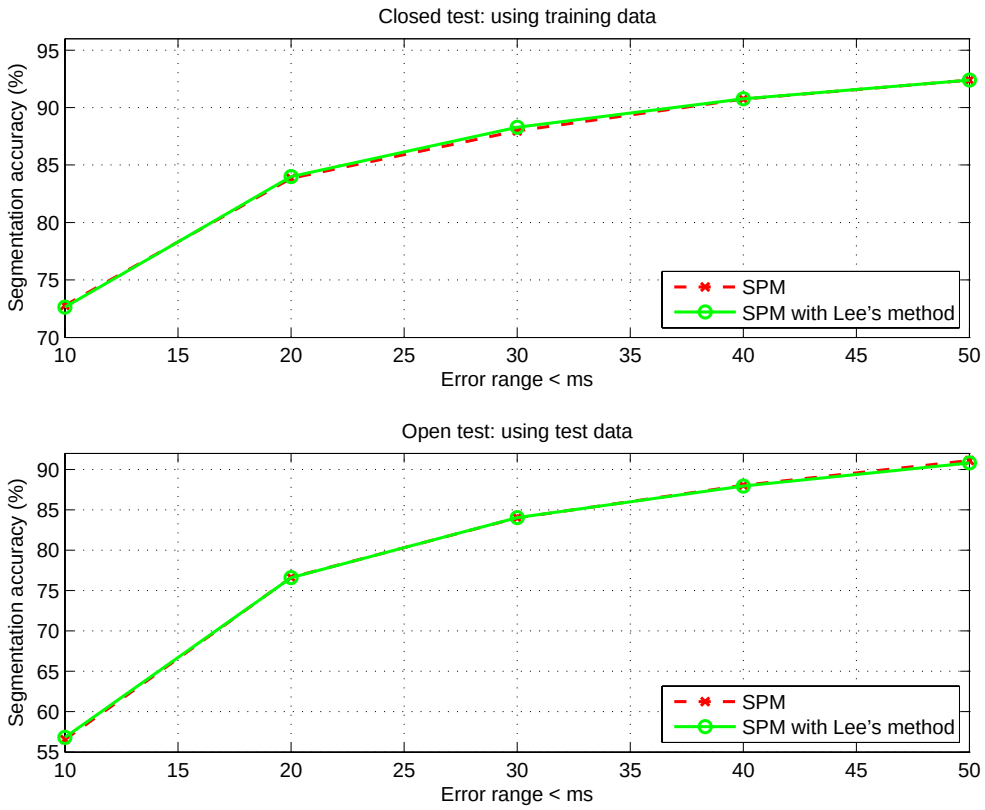


Fig. 6.16. Performance comparison of SPM and SPM combined with Lee's method. Top: closed test. Bottom: open test. (Evaluted data: SVS-1384)