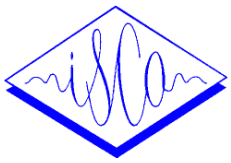

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

PREDICTING WORD SPOTTING PERFORMANCE

Man-hung Siu *Herbert. Gish* *Robin Rohlicek*
BBN Systems and Technologies
70 Fawcett Street, Cambridge MA 02138 USA

ABSTRACT

To use a word spotting system efficiently, it is helpful to be able to predict the performance of the system accurately. In this paper, we investigate performance prediction under different conditions. First, we discuss how to use statistical techniques to predict performance, and its variability on new unseen testing data. Second, we show that classification trees can be used to estimate the posterior probability of putative hits and that posterior probability can predict performance of unlabeled test data. Thirdly, we show that the classification tree method can generalize to predict spotting performance on new keywords.

1. INTRODUCTION

A word spotting system processes input speech waveforms and generates a set of triplets, each consists of a hypothesized keyword, the hypothesized time of occurrence and a score. Word spotting performance is reported using the Receiver Operating Characteristic (ROC) curve which plots detection rate against false alarm rate. Operating points of the spotter are selected on the basis of the ROC curves and the false alarm rate which depends on the requirement of different applications. In this article, several questions regarding word spotting scoring and performance prediction are investigated. First, how can ROC curves be estimated from limited testing data? Secondly, how stable or variable are the ROC curves? Thirdly, because word spotting performance is word dependent, how can spotting performance on new keywords be predicted?

In this paper, we take a statistical point of view on ROC curves and consider them as random curves which are estimated from test data. Several statistical techniques to predict word spotting performance on new data are proposed. In Section 2., methods for predicting performance on the new data with the same keyword set are investigated. In Section 3. we discuss ways to estimate performances on unlabeled test data. and in Section 4., we discuss ways to extend some techniques to predicting performance on new keywords. In Section 5., we report some preliminary experiments and results employing these techniques.

All experiments and analysis are performed based on results obtained using our large vocabulary phonetic word spotter described in [2].

2. PREDICTING PERFORMANCE ON NEW DATA

In this section, we try to address the issue of predicting the performance of a word spotter when it is run on new, similar quality data. Three different methods are investigated, namely,

1. cross-validation,
2. re-sampling, and
3. Monte Carlo simulation.

2.1. Cross-Validation

When an independent set of testing data is not available to evaluate word spotting performance, we propose using cross-validation. By cross-validation, we mean dividing the training data into N partitions and training the models using all but one of the partitions and testing on the left out partition. This process is repeated until each partition is used once for testing. If there is enough data in each run to give a meaningful ROC curve, cross-validation allows us to use the training data more efficiently and gives multiple measurement of system performance. Otherwise, we can combine the spotting output and compute a single ROC curve. In word spotting, one can train the acoustic and language models using N-1 partitions and test on the one partition left out.

2.2. Resampling

Resampling techniques, sometimes called bootstraping, are used to estimate data variability due to sampling. In this approach, no knowledge about the underlying distribution or family of distributions is assumed. Instead, the data is used to define an empirical distribution. Suppose N data samples are observed,

$$\vec{X} = (x_1, \ldots, x_N)$$

from which the empirical distribution $\tilde{P}$ can be defined. Let

$$\tilde{\theta} = \theta(x_1, \ldots, x_N)$$

be a measurement on the observations that we are interested in. Suppose we draw N independent samples, $x_1^*, \ldots, x_N^*$ are drawn from $\tilde{P}$, the empirical distribution and define,

$$\Theta = \theta(x_1^*, \ldots, x_N^*).$$

Θ is different from $\tilde{\theta}$ due to sampling variation. Estimating the distribution of Θ through repeated sampling of $\tilde{P}$ provide an estimates of the variability of $\tilde{\theta}$.

We now investigate how resampling can be applied to word spotting. In word spotting, our observed data is the set of true hits scores, false alarms scores and misses and the measurement we are interested in is the ROC curve. Suppose we have N putative hits (including true hits and false alarms) and M misses, and thus, we have N + M events. One way to estimate the variability of ROC curves is to sample N + M times with replacement from the data points. The problem with this approach is that we would have to assume that there are always the same number of events.

Alternatively, we can sample the time slices instead of the events. Assuming that we partition the time axis of length T into slices of length dt such that there is either an event or nothing in that interval and sample $n = T/dt$ times from these slices. The probability distribution for the number of events selected is binomial with $p = (N + M)/n$ and $n = T/dt$. Thus, the probability of selecting k events is given by

$$p(k) = \binom{n}{k} \frac{1}{n}^k \left(1 - \frac{1}{n}\right)^{n-k}$$

If we make the slices shorter and shorter, that is, when $dt \rightarrow 0$ and $n \rightarrow \infty$, then the Binomial distribution converges to a

Poisson distribution, $Poisson(\lambda = np = N + M)$. Using this approach, at each trial, we obtains a new set of hits and misses, $\tilde{L}$ according to a Poisson rate of $(N + M)$.

If each individual event is considered different, then, each is a sub-process and the number of times each event is selected is $Poisson(\lambda = 1)$ That is, the probability for a particular event to selected k times is given by,

$$p(k) \rightarrow \frac{1}{e}\frac{1}{k!}.$$

From each trial, we replicate each putative hit and miss according to $Poisson(\lambda = 1)$ and generate a new ROC curve. By running a large number of trials, we can estimate the distribution of ROC curves and estimate their confidence intervals.

2.3. Monte Carlo Simulation

Monte Carlo simulations are often used when the underlying distributions are too complicated to apply analytical methods. Our word spotting system represents keywords as a concatenated sequence of phonemes, where each phoneme is modeled by a Hidden Markov model. Using analytical methods to analyze the scores are very difficult and Monte Carlo simulation is used to obtain the distribution of true putative hit and false alarm scores when the data truly match the model. There are several advantages of using simulation. First, the underlying cause of errors independent of the match between model and data can be investigated when observation sequences are generated using training model. Second, comparing performance obtained using an independent generating model with performance obtained using real data, we can evaluate the goodness of our model assumption. Finally, simulation allows us to generate as much data as needed. From simulated observation sequences, we can also simulate the distribution of the ROC.

The following is the procedure we use to perform simulation.

1. Acoustic (phonetic) are trained to be used to generate observation.
2. Acoustic and language models are trained for the word spotter. This may be the same or different from the generation model.
3. Text used for testing is generated either by choosing some sentences from training or independent test data or generated from the language model. Each sentence is converted into a phoneme sequence.
4. Acoustic observations are generated by transversing the HMM model. At each state, based on the outputs of random number generator, we generate an observation and transition to next state.
5. Word spotting is performed on each simulated sentence.
6. Simulated word spotting results are evaluated.

3. PREDICT PERFORMANCE ON UNLABELED DATA

When we spot keywords on some new data where we do not know the truth (unlabeled data), it may be useful to have an estimate of how well the spotter performs.

In this section, we propose using a classification tree as an non-parametric estimator of the posterior probability of true hits. We show that the posterior probability is a better score for word spotting. And based on the posterior probability, we suggest ways to predict the ROC curves on unlabeled data.

3.1. Classification Trees

Instead of limiting ourselves to using the putative hit scores only, other features can be used to improve word spotting performance and prediction. One powerful capability of classification tree is the ability to combine different types of features, including real, ordered, categorical and binary. Classification trees can be viewed as being a non-parametric conditional probability estimators.

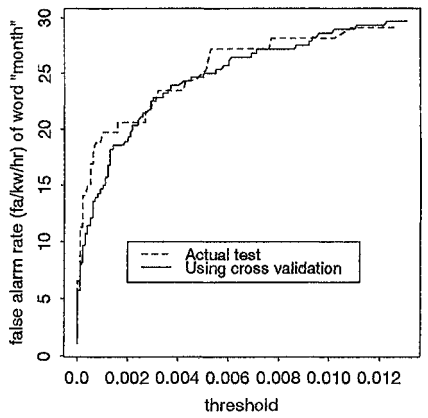

Figure 1. Comparing detection rate at different thresholds from cross validation experiment and real test

To use classification trees in word spotting, we consider all putative hits as two classes, the class of true hits and the class of false alarms. Each hit is associated with some features such as the HMM score, the number of phonemes in the keyword, the number of frame in detection. We use a greedy binary tree growing algorithm where at each node the algorithm selects a binary question on one feature such that the deviance is maximally reduced. After a tree is grown, it is pruned back using cross validation [1]. The goal of the tree is to classify putative hits correctly into the two classes. After the tree is built, putative hits are associated with different leaves where each leave consists of a class label and the estimated posterior probability of true hits.

3.2. Using Posterior Probability Scores

Traditionally, the word spotting operating point chosen on ROC curves drawn from some evaluation data is achieved by thresholding the putative hit scores. Typically, due to the word dependent nature of the score, these thresholds are set for each keyword independently. Frequently, these thresholds may be sensitive to a particular experiment. In Figure 1, we plot the false alarm rate against threshold from two different experiments on the keyword "month". As shown in this plot, the same threshold set on the cross validation experiment gives different false alarm rate on real data. Furthermore, a slight change in the threshold can change the detection dramatically.

Instead of setting the operating point based on the HMM scores, we can set thresholds on the posterior probabilities estimated by the classification tree. There are two advantages of using the posterior probabilities, 1) posterior probability has an absolute meaning and is more stable from experiment to experiment, 2) thresholds for multiple keywords can be set simultaneously. Figure 2 shows the tree score on the cross-validated experiment and the real test. Comparing Figure 2 to Figure 1, the posterior probabilities are more stable than the original HMM score for setting thresholds at the lower false alarm rates, which is the region of interest.

3.3. Using Posterior Probabilities to Predict ROC curves

Posterior probabilities can also be used to predict testing ROC curves. Because truth is unknown, we use the expected true hits and expected false alarms to plot ROC curves. Using the posterior probabilities, ROC curves can be estimated in the following

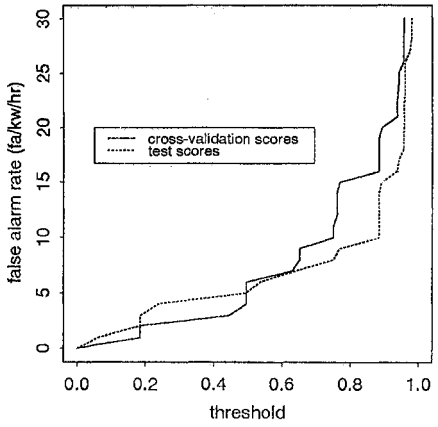

Figure 2. Comparing simulated performance with real test on some keywords

manner.

1. Rank order all hits using the tree score,
2. up to the nth best hit, the expected number of false alarms, is given by

$$fa(n) = \sum_{i<n} P_i(\text{fa})$$

and the expected detection rate is given by,

$$dt(n) = \frac{\sum_{i<n} P_i(\text{true hit})}{\sum_{j<N} P_j(\text{true hits})}$$

4. PREDICTING PERFORMANCE ON NEW KEYWORDS

Since performance between different keywords can vary significantly, it is not easy to interpolate new keyword performance from that of the other keywords. In this section, we describe how classification trees can be used to predict new keyword performance.

4.1. Classification Trees

Extending our method of predicting performance on new data using classification trees, we can predict new keyword performance. As mentioned earlier, classification trees can be trained using multiple features. If a classification tree is trained for many keywords, and the features we used sufficiently represent each keyword then the tree captures those factors that affect performance independent of the keywords. Whether the classification tree built is efficient in predicting the new keyword performance depends significantly on the features used and the number of keywords used in training. In effect, this approach is analogous to building a codebook of known keywords that we know and use the codebook to predict new keyword performance. If the different keywords used in training the tree sufficiently cover the feature space, then the performance of spotting new keywords can be predicted accurately based on this trained tree.

5. EXPERIMENTS

5.1. Background

Our experiments are done using the credit card task on switchboard database. Twenty keywords of different length and frequencies are selected. Training, including all work on cross-validation, is performed using 24 female conversation sides. Testing is performed using 11 conversation sides of independent female speakers. Each conversation side is about 5 minutes long. All experiments are performed using our large vocabulary word spotter where we use the backward forward posterior scores as reported in [3]. Since our goal is to understand the mechanics of the prediction process, a monophone phoneme loop alternate model is used which decreases our performance in general but simplifies our analysis.

Two kinds of ROC curves are referred to in our experimental results. They are composite and pooled ROC curves. A composite ROC curve is a weighted average of detection rates across different keywords at a fixed false alarm rate. A pooled ROC curve relates the total detection across all keywords at a common threshold and the total false alarm rate.

5.2. Experiments with Resampling and Cross Validation

We partitioned our training data into four approximatedly equal sets with no overlapping speakers to run cross validation. At each run, three parts were used for training and the fourth one was used for testing. This procedure was repeated 4 times and hits from all 4 runs were combined. The composite ROC curve of the combined set is shown in figure 3.

We performed a resampling experiment using the real test data. After spotting on the test data, the true hits, misses and false alarms were resampled 50 times to generate new composite ROC curves. Figure 3 shows the box plot of the resampled ROC curve on top of the cross-validated ROC curve and the real testing ROC. The solid line represents the median composite detection rate at a fixed false alarm rate and the boxes mark the range of the upper and lower quartile. We can see that the cross-validated ROC curve fell between the two quartiles.

5.3. Experiment with Simulation

Sentences from real test data were used as text for simulation. Alternatively, sentences can be generated using a language model. Acoustic data was generated using two different models. Model 1 was trained using the all training data and Model 2 was trained using all the test data. Our word spotter was trained using all training data. Spotting on data generated using Model 1 is similar to testing on training because of the perfect matched between data and model. Spotting on data generated on Model 2, however, should closely resemble spotting on fair testing data. Two experiments were performed. First, we compared the simulation results using Model 1 and Model 2 to investigate whether the training is sufficient. We believe with sufficient training, the bias of testing on training would be small. Second, we compared the simulation results from Model 2 with the real test to investigate the goodness of our model assumptions.

Figure 4 shows the composite ROC curves of spotting on data generated using both models as well as the test data. As shown by the curves, data generated with Model 1 performed better than data from Model 2. It is surprising that the composite ROC curve using Model 2 is still significantly better than the real test data. Our hypothesis is that the model for the data may not be very accurate.

5.4. Experiments with Classification Tree

A classification tree is built using labeled putative hits from 3 sets of cross-validated experiments. The classification tree is used to 1) estimate the posterior probabilities of the test data, 2) predict the ROC on testing data based on the posterior probabilities and, 3) predict new keyword performance.

Two different aspects of the estimated posterior probabilities are investigated. First, we compare its stability for thresholding the score as we discussed and shown in Figure 2. Then we compare the performance of pooled ROC curve generated using posterior probabilities and the original pooled ROC curve gener-

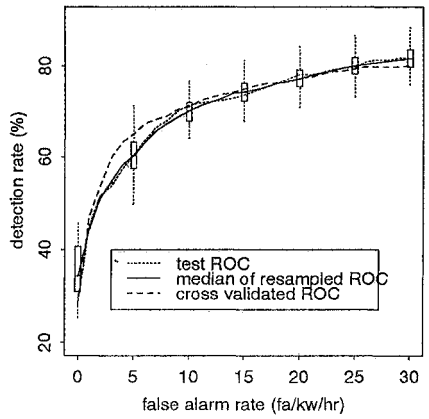

Figure 3. Resampling of the actual ROC

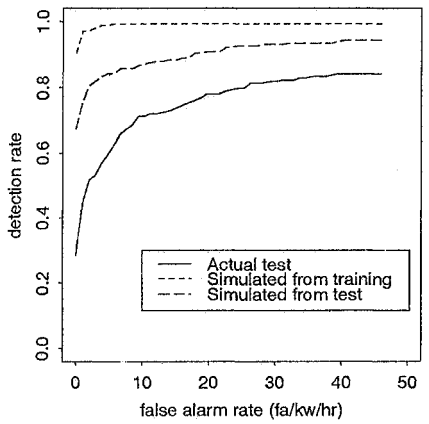

Figure 4. Comparing simulated performance with real test

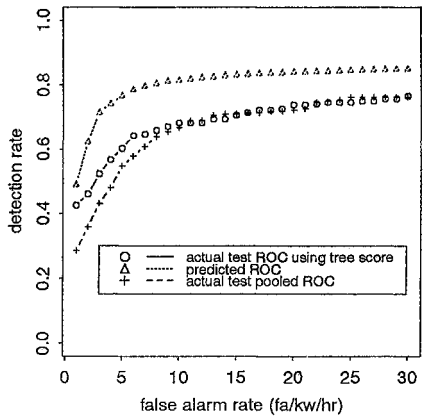

Figure 5. Comparing classification tree predicted ROC and actual ROC

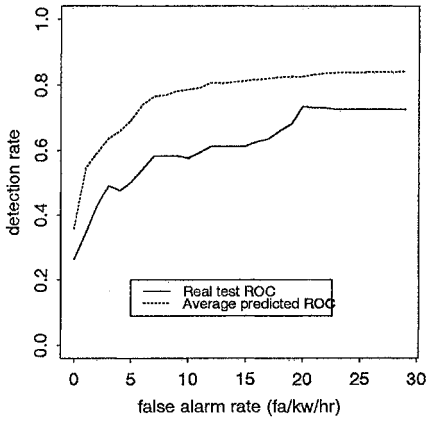

Figure 6. Comparing predicted new keyword performance with actual

ated from HMM score as shown in Figure 5. Finally, Figure 5 shows the predicted ROC curve based on the posterior probabilities. The predicted ROC curve is optimistically biased about 10% when compared to the actual ROC curve. This bias is probably due to insufficient training on the tree. The pooled ROC curve generated using posterior probability is slightly better than the original pooled ROC curve up to 10 false alarms per hour.

To predict new keyword performance, we perform leave-one-out experiment on the keywords. At each time, a tree was built using putative hits from 19 keywords in 3 sets of cross-validation experiments. Hits of the leave-out keywords in test were processed by the tree. Using the knowledge captured by the 19 training keywords, the tree predicted the performance of these new keywords. Figure 6 shows both the average ROC curve of the predicted performance and the actual performance using the posterior probabilities as scores. Similar to the prediction on new data, the tree's prediction is biased optimistically at about 15% in detection rate. We hypothesize better features and more keywords may help minimize this bias.

6. CONCLUSIONS

In this paper, we addressed the issue of predicting word spotting performance on new data, unlabeled test data and new keywords. We showed that cross-validation enables us to predict test performance accurately using only the training data only. We also showed that using resampling techniques, we can estimate confidence intervals for ROC curves. Then, we proposed using classification trees to estimate posterior probabilities on putative hits. Our experiments showed that although improvement on word spotting performance is modest, the method provides us with thresholds that give more consistent performance across experiments. We also showed how to use the posterior probability from classification trees to predict ROC curve on unlabeled test data, and how to predict performance on new keywords.

REFERENCES

[1] J. M. Chambers, T. J. Hastie "Statistical Models in S" Wadsworth & Brooks/Cole Advanced Books & Software, 1992 pp. 377-417.

[2] P. Jeanrenaud, M. Siu, K. Ng, R. Rohlicek, H. Gish, "Phonetic-based Word Spotter: Various Configurations and Application to Event Spotting." *Proc. ESCA Eurospeech*, 1993, vol. II, pp. 1057-1060

[3] R. Rohlicek, W. Russell, S. Roukos, H. Gish, "Continuous Hidden Markov Modeling for Speaker-Independent Word Spotting," 1989 IEEE ICASSP, pp.627-630.