BAYESIAN MODEL COMBINATION (BAYCOM) FOR IMPROVED RECOGNITION

Ananth Sankar

Nuance Communications, Menlo Park, CA 94025

ABSTRACT

It is well known that combining recognition outputs of multiple systems using methods such as ROVER or its extensions gives improved performance. However, previous approaches have been somewhat adhoc. In this paper, we present BAYCOM, a Bayesian decision-theoretic approach to model combination that is optimal under given assumptions. We present recognition experiments showing that BAYCOM gives significant improvements over previous combination methods. In addition, we show that BAYCOM provides a confidence feature that gives very large improvements over previous methods for utterance rejection.

1. INTRODUCTION

Since the introduction of the ROVER algorithm [1], model combination has been a popular way to improve recognition performance for automatic speech recognition (ASR) systems. Most state of the art research systems use ROVER, or related techniques like N-best ROVER [2] and Confusion Network Combination (CNC) [3] to combine models that have been trained in different ways. These methods use a plurality or confidence-weighted voting mechanism amongst a set of aligned word hypotheses derived from different systems. While they give good improvements, the combination approach used by these methods is adhoc and not grounded in a basic theory of pattern recognition.

In this paper, we motivate model combination from a Bayesian decision-theoretic viewpoint, and present an algorithm (BAYCOM) that optimally combines different models under certain commonly made assumptions. In addition to the recognition hypotheses, BAYCOM uses multiple scores from each system to make a final decision. BAYCOM makes no assumptions as to the meaning of these scores, unlike ROVER-based methods which require each system to provide a normalized confidence score. Also, unlike these previous methods, BAYCOM does not require that the individual models give similar error rates. Experiments show that BAYCOM is significantly superior to simple voting schemes, and also gives a confidence scoring mechanism that dramatically improves rejection performance.

2. BAYCOM: DECISION-THEORETIC MODEL COMBINATION

Suppose there are M models, each of which processes utterance x. Let the recognition hypothesis output by model i be $h_i(x)$. Further, model i outputs a set of L scores $s_i^j(x), j = 1, \ldots, L$ for hypothesis $h_i(x)$. For example, the scores may correspond to the confidence, normalized likelihood, or the likelihood difference between the top two hypotheses. The inputs to the model combination algorithm are $h_i(x)$, and $s_i^j(x)$, $i = 1, \ldots, M$, $j = 1, \ldots, L$.

We now proceed to frame the problem according to Bayes decision theory. Let the event h mean "hypothesis h is correct", and the set $H = \{h_1, \ldots, h_M\}$. Then we need to compute $h^*(x)$ such that

$$h^* = \underset{h \in H}{\operatorname{argmax}} P(h|h_1, \ldots, h_M, S_1, \ldots, S_M), \quad (1)$$

where we have removed the dependence on x for ease of presentation, and $S_i = s_i^1, \ldots, s_i^L$. We use Bayes theorem to write the probability term in Equation 1 as

$$P(h|h_1, \ldots, h_M, S_1, \ldots, S_M) = P(h)\frac{P(h_1, \ldots, h_M, S_1, \ldots, S_M|h)}{P(h_1, \ldots, h_M, S_1, \ldots, S_M)}$$

Ignoring, the denominator, which is independent of h, and assuming the model hypotheses are independent of each other, we substitute back in Equation 1 to get

$$h^* = \underset{h \in H}{\operatorname{argmax}} P(h) \prod_{i=1}^{M} P(S_i|h_i, h)P(h_i|h). \quad (2)$$

Consider the two disjoint subsets $I_C = \{i : h_i = h\}$ and $I_E = \{i : h_i \neq h\}$. Then

$$P(h_i|h) = \begin{cases} P_i(C) & \text{if } i \in I_C \\ P_i(E)/N - 1 & \text{if } i \in I_E \end{cases}$$

Here we have assumed that the probability of being correct, $P_i(C)$ for model i, is independent of the particular hypothesis h_i, and that the probability of error, $P_i(E) = 1 - P_i(C)$ is equally distributed over all the $N - 1$ incorrect hypotheses, where N is the number of possible unique hypotheses.

Also,

$$P(S_i|h_i,h) = \begin{cases} P(S_i|C) & \text{if } i \in I_C \\ P(S_i|E) & \text{if } i \in I_E \end{cases}$$

where $P(S_i|C)$, and $P(S_i|E)$ are the conditional score distributions given that the hypothesis h_i is correct and incorrect, respectively. We have assumed that the conditional score distributions are determined only by whether the hypothesis is correct or incorrect, and not by the actual hypothesis itself. We can now write the product term in Equation 2 as

$$\prod_{i=1}^{M} P(S_i|h_i,h)P(h_i|h) = \prod_{i\in I_C} P_i(C)P(S_i|C) \prod_{i\in I_E} \frac{P_i(E)}{N-1} P(S_i|E)$$

Multiplying and dividing by $\Pi_{i=1}^{M} \frac{P_i(E)}{N-1} P(S_i|E)$, we get

$$\prod_{i=1}^{M} P(S_i|h_i,h)P(h_i|h) = \prod_{i\in I_C} (N-1)\frac{P_i(C)}{P_i(E)} \frac{P(S_i|C)}{P(S_i|E)} \prod_{i=1}^{M} \frac{P_i(E)}{N-1} P(S_i|E)$$

Realising that the second term $\prod_{i=1}^{M} \frac{P_i(E)}{N-1} P(S_i|E)$ is a constant, we substitute back in Equation 2 to get

$$h^* = \underset{h\in H}{\operatorname{argmax}} P(h) \prod_{i:h_i=h} (N-1)\frac{P_i(C)}{P_i(E)} \frac{P(S_i|C)}{P(S_i|E)} \quad (3)$$

Taking the logarithm of the right hand side, we get

$$h^* = \underset{h\in H}{\operatorname{argmax}} \begin{bmatrix} \log P(h) \\ + \sum_{i:h_i=h} \log \frac{P_i(C)(N-1)}{P_i(E)} \\ + \sum_{i:h_i=h} \log \frac{P(S_i|C)}{P(S_i|E)} \end{bmatrix} \quad (4)$$

Equation 4 gives the model combination formula for BAYCOM. We note several of its desirable properties:

- BAYCOM is a weighted voting scheme with voting weights given by $\log \frac{P_i(C)(N-1)}{P_i(E)} + \log \frac{P(S_i|C)}{P(S_i|E)}$
- Consider the term $\log \frac{P_i(C)(N-1)}{P_i(E)}$. If a model is always *correct*, i.e., $P_i(C) = 1, P_i(E) = 0$, then its vote is weighted by ∞.
- If a model is always *incorrect*, i.e., $P_i(C) = 0$, then its vote is weighted by $-\infty$.
- If a model is *simply guessing*, i.e., $P_i(C) = \frac{1}{N}$, then its vote is not counted (weight is 0).
- Similar observations can be made about the conditional distributions, $P(S_i|C)$, and $P(S_i|E)$.
- A tied vote is broken by choosing the hypothesis with the maximum prior, $P(h)$.

Finally, to mitigate the model independence assumptions we made, we can multiply the weighting terms by a model-specific weight, α_i, so that

$$h^* = \underset{h\in H}{\operatorname{argmax}} \begin{bmatrix} \log P(h) \\ + \sum_{i:h_i=h} \alpha_i \log \frac{P_i(C)(N-1)}{P_i(E)} \\ + \sum_{i:h_i=h} \alpha_i \log \frac{P(S_i|C)}{P(S_i|E)} \end{bmatrix} \quad (5)$$

This weighting is similar in effect to the grammar probability weighting used in speech recognition.

In our experiments, each model outputs a semantic hypothesis which corresponds to a meaningful action that the system can take. For example, "Charles Schwab" may be the semantic hypothesis corresponding to the sentence hypothesis, "I'd like to connect to Schwab please". Voting in our experiments is on these semantic actions. BAYCOM can easily be extended for improving word error by replacing the voting procedures of ROVER or CNC with that of BAYCOM. It is also possible to implement BAYCOM at lower levels, such as the phone level.

3. RELATION TO OTHER METHODS

BAYCOM relates to ROVER [1], N-best ROVER [2], and CNC [3] in that all are voting mechanisms. While ROVER combines only a single hypothesis from each model, N-best ROVER and CNC combine multiple hypotheses represented as confusion networks. However, all use a simple confidence weighted voting scheme. By contrast, BAYCOM derives the optimal combination weight from decision theory. It properly handles multiple scores of different types from each model. It makes no assumption as to the meaning of these scores, instead using the conditional score distributions to properly weight the votes. ROVER and CNC, on the other hand, require individual systems to give a confidence score in the $[0-1]$ range indicating zero to total confidence in the recognition hypothesis. Further, BAYCOM does not require that the individual models have similar error rates, as opposed to ROVER or CNC, which typically work well when the individual models have similar error rates.

BAYCOM is also similar to boosting methods such as AdaBoost [4]. However, boosting uses only the probability of being correct for each model, whereas our approach also optimally uses multiple model scores $S_i = s_i^1, \ldots, s_i^L$ for hypothesis h_i.

4. A POWERFUL NEW CONFIDENCE SCORE

Intuitively, one would expect model combination to also provide improved confidence scores. For example, if a hypothesis receives a vote from several models, we may want to assign it a high confidence. A reasonable measure of this confidence is the posterior probability of the hypothesis given the information contributed by the various models. BAYCOM provides a natural framework for computing such a score. From our development in Section 2, we can write this score as

$$c(h^*) = P(h^*|h_1, \ldots, h_M, S_1, \ldots, S_M),$$

or

$$\begin{aligned} c(h^*) &= \log P(h^*) \\ &\quad + \sum_{i:h_i=h^*} \alpha_i \log \frac{P_i(C)(N-1)}{P_i(E)} \\ &\quad + \sum_{i:h_i=h^*} \alpha_i \log \frac{P(S_i|C)}{P(S_i|E)} \end{aligned} \tag{6}$$

5. EXPERIMENTAL RESULTS

We conduct experiments on a large directory assistance (DA) task. About 750,000 utterances from the domain are used to train a set of gender-independent (GI) task-specific Genone-based hidden Markov models (HMMs) with 2000 Genones. A Genone is the set of Gaussians corresponding to a state cluster [5]. This is our baseline acoustic model. Two other models are trained on a much larger task-independent dataset of about 6M utterances that does not include these 750,000 utterances. The first of these is a system that generates n-best lists using a 1000 Genone GI system, and rescores these lists using a 2000 Genone gender dependent (GD) model selected for each utterance based on a gender classifier. The second model is a 2000 Genone GI system. All models use 32 Gaussians per Genone. We refer to the three models as

Baseline : The task-specific acoustic model with 2000 GI Genones

gd1000 : The task-independent acoustic model with 1000 GI Genones and rescoring with 2000 GD Genones

gi2000 : The task-independent acoustic model with 2000 GI Genones

The BAYCOM parameters, $P_i(C)$, $P(S_i|C)$, and $P(S_i|E)$ are trained on a 380,000 in-grammar (IG) subset of the 750,000 utterance training set. The IG set is necessary to run recognition experiments to estimate $P_i(C)$, $P(S_i|C)$, and $P(S_i|E)$. We used a single utterance-dependent score $S_i = s_i^1$ in our experiments. This was the confidence score computed by the Nuance recognition system for each utterance [6]. We modeled $P(s_i^1|C)$, and $P(s_i^1|E)$ using normalized histograms with 10 bins.

For testing, we used about 50,000 utterances from the DA task. Our error metric counts the number of semantic hypothesis errors.

5.1. Comparison to previous voting schemes

We compared BAYCOM to two common voting approaches. The first is simple plurality voting, and the second weights each vote by the raw utterance confidence score. Note that BAYCOM weights the vote by the logarithm of the ratio of the confidence score distributions as opposed to the raw score itself. We also evaluate the effect of adding the prior term $logP(h)$, and the model specific weights, α_i in the voting formula of Equation 5. The model weights α_i are trained so as to minimize the semantic error on the 380,000 utterance training set. There are about 14000 unique semantic classes with a perplexity of 953, so we used a value of $N = 1000$ in our experiments. Table 1 shows that BAYCOM is significantly better than either simple plurality voting or confidence-weighted voting. In fact, both these methods were worse than the baseline system, perhaps because the individual systems had widely varying error-rates. However, this did not pose a problem for BAYCOM. When we utilized the prior and model-specific weights, BAYCOM was 13% better than the baseline.

Individual Models	
Baseline	4.48%
gd1000	7.12%
gi2000	8.58%
Combination of above models	
Plurality voting	5.87%
Confidence-weighted voting	4.71%
BAYCOM with no hypothesis prior	4.35%
BAYCOM with hypothesis prior $logP(h)$	4.03%
BAYCOM with prior and model weights α_i	3.91%

Table 1. BAYCOM compared to standard approaches

5.2. Diminishing returns with more models

To evaluate the gains from adding more than one model to the baseline, we studied the effect of adding *gd1000* and *gi2000* in steps. BAYCOM with priors and model weights is used, and the results are given in Table 2. We see that most of the gain is achieved by adding a single model, and only a minor additional gain is gotten from an additional

Model	Error Rate	Relative Improvement
Baseline	4.48%	
+ gd1000	3.95%	12%
+ gi2000	3.91%	13%

Table 2. Effect of adding models in steps for BAYCOM

model. This result is likely to vary with the choice of individual models. We have made no attempt to train independent individual models, and have focussed instead on the problem of combination. Boosting methods such as AdaBoost [4] do train the models in a way to achieve more independent errors, and it would be interesting to incorporate those schemes with the more general combination approach presented in this paper.

5.3. Improvements in confidence and rejection

As we noted in Section 4, BAYCOM provides a potentially powerful confidence score. We evaluated this by computing the confidence score using Equation 6 for the model combination schemes studied in Section 5.2. The baseline system used the Nuance system's standard confidence score computed as in [6]. Confidence score performance was compared by plotting an ROC curve of the *miss rate* against the *false accept rate* using both in-grammar (IG) and out-of-grammar (OOG) test data. An IG utterance is one that can be parsed by the application grammar, and an OOG utterance is one that cannot be. A *false accept* occurs either when an IG utterance is incorrectly recognized, or an OOG utterance is accepted by the system. A *miss* occurs when an IG utterance is incorrectly recognized or is rejected. The *miss rate* is the same as $1 - \textit{correct automation rate}$. Improving *correct automation* at low *false accepts* is an important measure in applications like directory assistance.

We simulated a 10% OOG rate, and plotted the corresponding ROC curve in Figure 1 for the baseline model and the two BAYCOM schemes we evaluated in Section 5.2. This figure shows that BAYCOM gives very large improvements in the *miss rate* at low *false accept rates*. For example, we get a 43% improvement in the *miss rate* at *1% false accept rate*.

6. CONCLUSION

We presented a Bayesian decision-theoretic approach to model combination (BAYCOM), and described its relationship to existing approaches. We conducted comprehensive experimental studies on a directory assistance task, showing that BAYCOM was significantly better than standard voting schemes such as plurality voting or confidence-weighted

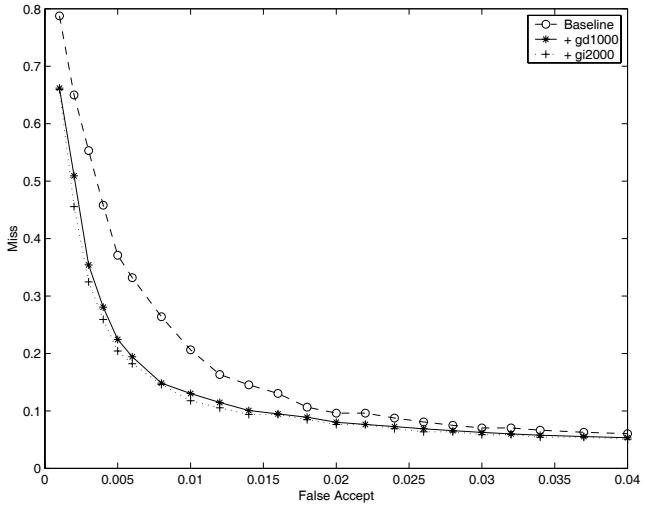

Fig. 1. Miss rate vs. False Accept rate for 10% OOG

voting, and gave a 13% improvement over the baseline task-specific model. The BAYCOM confidence scoring mechanism gave a 43% improvement in the *miss rate* at a 1% *false accept rate*. In summary, BAYCOM is elegant, theoretically interesting, and gives very attractive gains.

7. REFERENCES

[1] J.G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. ASRU*, December 1997, pp. 347–354.

[2] A. Stolcke, et al., "The SRI March 2000 Hub-5 Conversational Speech Transcription System," in *Proc. Speech Transcription Workshop*, 2000.

[3] G. Evermann and P.C. Woodland, "Posterior Probability Decoding, Confidence Estimation, and System Combination," in *Proc. Speech Transcription Workshop*, 2000.

[4] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Comp. and Sys. Sciences*, vol. 55, no. 1, pp. 119–139, August 1997.

[5] V. Digalakis, P. Monaco, and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 4, pp. 281–289, 1996.

[6] E. Chang, "Improving Rejection with Semantic Slot-Based Confidence Scores," in *Proceedings of EUROSPEECH*, 1999, pp. 271–274.