Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An iterative approach to Bayes risk decoding and system combination

Hai-hua XU, Jie ZHU

(*Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

E-mail: {haihua_xu, zhujie}@sjtu.edu.cn

Received Mar. 3, 2010; Revision accepted July 6, 2010; Crosschecked Jan. 31, 2011

Abstract: We describe a novel approach to Bayes risk (BR) decoding for speech recognition, in which we attempt to find the hypothesis that minimizes an estimate of the BR with regard to the minimum word error (MWE) metric. To achieve this, we propose improved forward and backward algorithms on the lattices and the whole procedure is optimized recursively. The remarkable characteristics of the proposed approach are that the optimization procedure is expectation-maximization (EM) like and the formation of the updated result is similar to that obtained with the confusion network (CN) decoding method. Experimental results indicated that the proposed method leads to an error reduction for both lattice rescoring and lattice-based system combinations, compared with CN decoding, confusion network combination (CNC), and ROVER methods.

Key words: Bayes risk (BR), Confusion network, Speech recognition, Lattice rescoring, System combination
doi:10.1631/jzus.C1000045 **Document code:** A **CLC number:** TP391.42

1 Introduction

1.1 Problem overview

Bayes risk (BR) decoding has been a research focus in the speech community for years. It is more flexible in application, but difficult to implement compared with the popular maximum a posteriori (MAP) decoding scheme. To better understand the BR decoding principle, we first present the expected risk criterion:

$$\mathcal{H}(R)=\sum_{\mathcal{W}}P(\mathcal{W}|\mathcal{O})\mathcal{L}(\mathcal{W},R), \tag{1}$$

where R is a given hypothesis sequence, $\mathcal{W}$ is one of the many hypothesis sequences corresponding to the acoustic observation sequence $\mathcal{O}$, and $P(\mathcal{W}|\mathcal{O})$ is the posterior probability. $\mathcal{L}(\mathcal{W},R)$ represents a loss criterion measuring mismatch between $\mathcal{W}$ and R according to a specific loss definition, and $\sum$ is normally represented as an N-best sentence list or a word graph (lattice) in speech recognition.

With the expected risk definition in Eq. (1), the BR decoding rule can be represented as

$$\begin{aligned} R^* &= \text{argmin}_R\mathcal{H}(R) \\ &= \text{argmin}_R\sum_{\mathcal{W}}P(\mathcal{W}|\mathcal{O})\mathcal{L}(\mathcal{W},\mathcal{R}), \end{aligned} \tag{2}$$

where if we define $\mathcal{L}(\mathcal{W},R)$ on sentence level, that is,

$$\mathcal{L}(\mathcal{W},R)=\begin{cases}1, & \text{if } \mathcal{W}{=}R,\\ 0, & \text{otherwise,}\end{cases} \tag{3}$$

then the BR decoding rule is implemented as the standard MAP decoding rule, which actually gives the BR estimate with respect to the minimum sentence error. However, the sentence error is not a metric by which speech recognition systems are usually evaluated; we generally use a word error rate (WER) based on the Levenshtein edit distance (Levenshtein, 1966). A natural solution to this problem

is to use the BR estimate with respect to the minimum word error (MWE) in Eq. (2), where $\mathcal{L}(\mathcal{W}, R)$ is defined on word level:

$$\mathcal{L}(w, r) = \begin{cases} 0, & \text{if } w{=}r, \\ 1, & \text{otherwise}, \end{cases} \tag{4}$$

where $w \in \mathcal{W}$ and $r \in R$ are two words, and $\mathcal{L}(\mathcal{W}, R)$ is calculated as the Levenshtein edit distance between word sequences $\mathcal{W}$ and R.

1.2 Difficulty of computing the Bayes risk criterion

It is too prohibitive to do the computation implied in Eq. (2) concerning the MWE. Using a lattice, we would need two nested loops over word sequence $\mathcal{W}$, one for the argmin and the other for the $\sum$, both of which would be ranged over a huge number of word sequences in the lattice. Additionally, the inner part, where we compute $\mathcal{L}(\mathcal{W}, R)$, generally requires computations with $O(|\mathcal{W}||R|)$ time complexity. Taking all these into account, an approximate method should be employed.

1.3 Previous work with Bayes risk decoding in the minimum word error

The first paper to introduce the concept of BR decoding in terms of MWE optimization was Stolcke *et al.* (1997), where the $\sum$ was approximated by a long N-best list (about 1000 sentences), the argmin was approximated by a short N-best list (e.g., 10), and $\mathcal{L}(\mathcal{W}, R)$ was computed using a standard quadratic-time algorithm. This paper was a proof of the concept but it was not practical. Soon afterwards, two different workable approximations were proposed (Goel *et al.*, 2000; Mangu *et al.*, 2000), and one of them is confusion network (CN) decoding, which has become a widely applied approach. In particular, it is successfully applied to system combination, known as confusion network combination (CNC) (Evermann and Woodland, 2000), which improves on the one-best decoding combination method known as ROVER (Fiscus, 1997). Following the work in Goel *et al.* (2000), a lattice segmentation approach to BR decoding was proposed by Goel *et al.* (2004); experimental results showed that it outperforms the MAP method in terms of WER.

We also note some more recent work. Wessel *et al.* (2001) exactly optimized a frame-by-frame word correctness criterion over a lattice, and it was shown that the time-frame error rate is correlated with the actual WER. HakKani-Tur and Riccardi (2003) used a pivot to chunk the lattice, leading to results comparable to those with the CN method. Our previous work Xu *et al.* (2009a) used similar approximations as used in MPE/MWE (minimum phone error/minimum word error) training (Povey and Woodland, 2002) to address the summation in Eq. (2). This is also similar to one of the approaches described in Hoffmeister *et al.* (2006). Another interesting piece of related work is Heigold *et al.* (2005), applied to the MPE/MWE training using a weighted finite state transducer (WFST).

1.4 Our approach

In this work, we propose a new approximation approach to BR decoding that focuses on MWE optimization. Our approach is an iterative forward-backward algorithm combined with the dynamic-programming (DP) edit-distance computation on the lattice. The entire approach is composed of three procedures as the following:

1. Run forward algorithm on the lattice. To achieve this, a hypothesis reference R is first introduced, and then lattice-based DP is performed; as a result, the approximated value of the expected error in Eq. (1) is yielded.

2. Run backward algorithm on the lattice to accumulate a statistics $\gamma(k, w)$, which describes a likelihood of aligning word label w in the lattice to position k in R, $1 \le k \le |R|$.

3. Update R with the accumulated statistics $\gamma(k, w)$ iteratively till the estimated expected risk is not changed any more.

The overall procedure is similar to a discrete expectation-maximization (EM) algorithm implementation, where each time we update the hypothesis reference it can be guaranteed that the approximated expected risk in Eq. (1) will decrease. In this work, we focus the application of the proposed approach on two tasks: one is for single lattice rescoring; the other is for lattice-based system combination decoding. Based on our previous work in Xu *et al.* (2010), we will elaborate the proposed approach in a different scenario.

2 Iterative Bayes risk decoding

2.1 Lattice notations

In speech recognition, the word 'lattice' describes a structure that stores a set of alternative transcriptions of a hypothesis utterance. A lattice $\mathcal{T}$ is essentially a directed acyclic graph composed of a set of nodes and arcs, with a unique beginning node $b(\mathcal{T})$ and ending node $e(\mathcal{T})$. The former has an outgoing arc set represented as $\text{foll}(b(\mathcal{T}))$, and the latter has an incoming arc set represented as $\text{pred}(b(\mathcal{T}))$, while for the other non-endpoint node $n \in \text{node}(\mathcal{T})$, each has two connected arc sets, represented as $\text{pred}(n)$ and $\text{foll}(n)$. Each arc $a \in \text{arc}(\mathcal{T})$ has such information as a starting node $s(a)$, $a \in \text{foll}(s(a))$, and ending nod $e(a)$, $a \in \text{pred}(e(a))$, word label $w(a) \in \mathcal{S}$, where $\mathcal{S}$ is the whole word label set with $\mathcal{T}$, and the arc transitional cost $p(a)$ that is generally calculated as

$$p(a) = \left(p(\mathcal{O}|w(a))p(w(a)|h)^{\kappa_l}\right)^{\kappa}, \tag{5}$$

where $p(\mathcal{O}|w(a))$ is the acoustic score, $p(w(a)|h)$ is the language model score, and κ_l and κ are the language scaling factor and lattice scaling factor, respectively. To do BR decoding, we typically scale down the acoustic probability, that is, $\kappa < 1.0$, and normally $\kappa = 1/\kappa_l$.

2.2 Statistic quantity definition

To familiarize readers with the statistic quantities employed in the following algorithms, we explain them in Table 1.

2.3 Forward algorithm for expected risk

We address the problem of calculating an approximated expected risk of Eq. (1) with an improved forward algorithm given a hypothesis reference R. The novelty of the algorithm lies in the fact that it is combined with the DP computation between the objective lattice $\mathcal{T}$ and a hypothesis reference R. Theoretically, it is a closer approximation to the real solution of Eq. (1), in contrast with our previous work in Xu *et al.* (2009a), since the loss criterion $\mathcal{L}(\cdot,\cdot)$ calculation is globally optimized in this work. The procedure is illustrated in Algorithm 1.

The main goal of Algorithm 1 is to obtain an estimate of the expected risk $E = \hat{\mathcal{H}}(R)$ of the criterion Eq. (1), given a lattice $\mathcal{T}$ and a hypothesis reference R. The key quantity of the algorithm is the forward average error $\alpha'(n,k)$, which represents a weighted edit distance for node n being aligned to position k ($1 \le k \le |R|$) of R. We notice that each $\alpha'(n,k)$ calculation is associated with one of the three categories of errors given correctness as a special case of the substitution error, and $b(n,k)$ quantity is only a pointer that marks a deletion error, which will be used to facilitate the backward algorithm implementation. Table 2 lists error types versus corresponding representations of Algorithm 1.

Table 1 Definition of the key quantities

Notation	Definition
$\alpha(n)$	Forward probability from the start node $b(\mathcal{T})$ up to node n, which is recursively calculated as $\alpha(n) = \begin{cases} 1, & n = b(\mathcal{T}), \\ \sum_{a\in\text{pred}(n)} \alpha(s(a))p(a), & n \neq b(\mathcal{T}). \end{cases}$
$\alpha'(n,k)$	Forward expected error estimate with node n being aligned to position k ($1 \le k \le \lvert R\rvert$) of the hypothesis reference R
$\beta(n,k)$	Backward probability leading from node n to the final node $e(\mathcal{T})$, calculated as $\begin{cases} \text{Deletion:} & \beta(n,k-1) \mathrel{+}= \beta(n,k), \\ \text{Substitution:} & \beta(s(a),k-1) \mathrel{+}= \beta(n,k)p(a), \\ \text{Insertion:} & \beta(s(a),k) \mathrel{+}= \beta(n,k)p(a), \end{cases}$ where $a \in \text{pred}(n)$, and $\beta(n,k)$ is recursively derived according to the alignment error between hypothesis reference and lattice
$\gamma(k,w)$	The accumulated probability of alignments of a class of word arcs that are labeled with w and aligned to position k ($1 \le k \le \lvert R\rvert$) of the hypothesis reference R. Normally $\gamma(k,w) = \sum_{a\in\mathbf{arc}(\mathcal{T}):w(a)=w} \frac{\alpha(s(a))p(a)\beta(e(a),k)}{\alpha(e(\mathcal{T}))}$

Table 2 Representation of alignment errors

Error type	Representation
Deletion	$b(n,k) = \text{true}$
Substitution	$S \le I$
Insertion	$S > I$

$S = \alpha'(s(a),k-1) + \mathcal{L}(w(a),r_k)$, $I = \alpha'(s(a),k) + \mathcal{L}(w(a),\epsilon)$, $a \in \text{pred}(n)$, $b(n,k) = \text{false}$

2.4 Backward algorithm

Based on the error type for each pair of alignment from the forward algorithm, a two-dimensional backward algorithm is performed to accumulate statistics $\gamma(k,w(a))$, where $w(a)$ is a word label of arc a, k ($1 \le k \le |R|$) refers to a position of the hypothesis reference R. Algorithm 2 illustrates the detailed procedure of the algorithm.

Algorithm 1 Forward algorithm for expected risk

1: $\mathcal{T} \leftarrow \text{TopSort}(\mathcal{T})$ // Topologically sort lattice $\mathcal{T}$
2: $N \leftarrow |\mathcal{T}|$, $K \leftarrow |R|$
3: $\forall(n,k),\ b(n,k) = \text{false}$
4: $\alpha(1) = 1,\ \alpha'(1,0) \leftarrow 0$
5: **for** $k \leftarrow 1$ to K **do**
6: $\quad \alpha'(1,k) \leftarrow \alpha'(1,k-1) + \mathcal{L}(\epsilon, r_k)$
7: $\quad b(1,k) \leftarrow \text{true}$
8: **end for**
9: **for** $n \leftarrow 2$ to N **do**
10: $\quad \alpha(n) \leftarrow \sum_{a \in \text{pred}(n)} \alpha(s(a))p(a)$
11: $\quad$ **for** $a \in \text{pred}(n)$ **do**
12: $\quad\quad p \leftarrow \alpha(s(a))p(a)/\alpha(n)$
13: $\quad\quad$ **for** $k \leftarrow 0$ to K **do**
14: $\quad\quad\quad t = p\min \left\{ \begin{array}{c} \alpha'(s(a),k-1) + \mathcal{L}(w(a), r_k) \\ \alpha'(s(a),k) + \mathcal{L}(w(a), \epsilon) \end{array} \right\}$
15: $\quad\quad\quad \alpha'(n,k) \mathrel{+}= t$
16: $\quad\quad$ **end for**
17: $\quad$ **end for**
18: $\quad$ **for** $k \leftarrow 1$ to K **do**
19: $\quad\quad$ **if** $\alpha'(n,k) > \alpha'(n,k-1) + \mathcal{L}(\epsilon, r_k)$ **then**
20: $\quad\quad\quad \alpha'(n,k) = \alpha'(n,k-1) + \mathcal{L}(\epsilon, r_k)$
21: $\quad\quad\quad b(n,k) = \text{true}$
22: $\quad\quad$ **end if**
23: $\quad$ **end for**
24: **end for**
25: $P \leftarrow \alpha(N)$
26: $E \leftarrow \alpha'(N,K)$ // Approx. expected error given R

Algorithm 2 Backward algorithm illustration

1: $\forall(n,k),\ \beta(n,k) \leftarrow 0$ // Initialization
2: $\forall(k,a),\ \gamma(k,a) \leftarrow 0$
3: $\beta(N,K) \leftarrow 1$
4: **for** $n \leftarrow N$ to 1 **do**
5: $\quad$ **for** $k \leftarrow K$ to 0 **do**
6: $\quad\quad$ **if** $b(n,k)$ **then**
7: $\quad\quad\quad \gamma(k,\epsilon) \leftarrow \gamma(k,\epsilon) + \alpha(n)\beta(n,k)/P$
8: $\quad\quad\quad \beta(n,k-1) \leftarrow \beta(n,k-1) + \beta(n,k)$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad$ **for** $a \in \text{pred}(n)$ **do**
12: $\quad\quad$ **for** $k \leftarrow 0$ to K **do**
13: $\quad\quad\quad S = \alpha'(s(a),k-1) + \mathcal{L}(w(a), r_k)$
14: $\quad\quad\quad I = \alpha'(s(a),k) + \mathcal{L}(w(a), \epsilon)$
15: $\quad\quad\quad$ **if** $\neg b(n,k)$ **then**
16: $\quad\quad\quad\quad$ **if** $(k > 0) \wedge (S \leq I)$ **then**
17: $\quad\quad\quad\quad\quad \gamma(k, w(a)) \mathrel{+}= \alpha(s(a))\beta(n,k)p(a)/P$
18: $\quad\quad\quad\quad\quad \beta(s(a),k-1) \mathrel{+}= \beta(n,k)p(a)$
19: $\quad\quad\quad\quad$ **else**
20: $\quad\quad\quad\quad\quad \beta(s(a),k) \leftarrow \beta(s(a),k) + \beta(n,k)p(a)$
21: $\quad\quad\quad\quad$ **end if**
22: $\quad\quad\quad$ **end if**
23: $\quad\quad$ **end for**
24: $\quad$ **end for**
25: **end for**

Algorithm 2 indicates that the backward algorithm is conducted according to the error types as shown in Table 2. For $\beta(n,k)$ recursion, when a deletion occurs, do line 8; when a substitution occurs, do line 18; finally, do line 20 for an insertion error. For $\gamma(k, w(a))$ accumulation, it can be explained as follows: given a position k ($1 \leq k \leq |R|$), when an arc (or word label) is aligned to k with substitution, we accumulate $\gamma(k, w(a))$ with line 17; when an arc is aligned to k with deletion, we accumulate $\gamma(k, \epsilon)$ with line 7, where ϵ represents an empty word. However, when it is an insertion error, the algorithm treats no accumulation for γ quantity. We do this to prevent more than one word of a sentence being aligned to the same position of R. To this end, we normalize the original R; that is, if we write the original R as $(r_1 r_2 \dots r_{|R|})$, where r_i ($1 \leq i \leq |R|$) is a word, we turn it into $(\epsilon r_1 \epsilon r_2 \dots r_{|R|} \epsilon)$, which implies that each normal word r in R is surrounded by an empty word ϵ. We do this to deal with the situation in which those words having insertion errors with the original R can be aligned to an ϵ word, which keeps an important feature of the quantity $\gamma(k, w(a))$ as follows:

$$\sum_{a \in \text{arc}(\mathcal{T})} \gamma(k, w(a)) = 1, \text{ for } 1 \leq k \leq |R|. \quad (6)$$

Eq. (6) can be explained in a simplified way. As analyzed, each path in the lattice has one and only one alignment with each position of the hypothesis reference, which is justified with Algorithm 1. If there are M paths in the lattice and the posteriori probability of each path is p_i ($1 \leq i \leq M$), then we have $\sum_{i=1}^{M} p_i = 1$. Each position k of R ($1 \leq k \leq |R|$) is aligned by each path with probability p_i. Therefore, Eq. (6) is established.

2.5 Hypothesis reference update

When $\gamma(k, w(a))$ is accumulated, we select those symbols for each position k ($1 \leq k \leq |R|$) to update R such that $\hat{r}_k = \max_{w(a)} \gamma(k, w(a))$. We do this until E does not converge. For clarity, the entire procedure is illustrated in Algorithm 3.

Four points in Algorithm 3 are worth emphasizing. One is that we take the MAP decoding result to initialize, as we think this can improve the final

Algorithm 3 Iterative algorithm for Bayes risk decoding

1: $R' \leftarrow$ MAP-Decode($\mathcal{T}$) // One-best to initialize
2: $R' \leftarrow$ AddEps(R') // Add ϵ word to normalize R'
3: **loop**
4: do forward algorithm, yielding estimate $\hat{E}_1$
5: **for** $k \leftarrow 1$ to $|R'|$ **do**
6: do backward algorithm, obtaining $\gamma(k, w(a))$
7: update to R: $r_k \leftarrow \max_{w(a)} \gamma_{R'}(k, w(a))$
8: **end for**
9: do line 4 with R, obtaining $\hat{E}_2$
10: **if** $\hat{E}_2 - \hat{E}_1 = 0$ **then**
11: break
12: **end if**
13: $R' \leftarrow R$
14: $\hat{E}_1 \leftarrow \hat{E}_2$
15: $R' \leftarrow$ AddEps(R')
16: **end loop**
17: $R' \leftarrow$ RemoveEps(R') // Remove ϵ word to output

result and lead to a rapid convergence. The second is that an ϵ word must be added to R to solve insertion problems before each iteration. The third is that the current estimate $\hat{E}_2$ must be no larger than the last estimate $\hat{E}_1$, i.e., $\hat{E}_2 \leq \hat{E}_1$, and if $\hat{E}_2 = \hat{E}_1$, the iteration terminates. The last point is that we must remove the ϵ word from R before taking it as the decoding result.

2.6 Algorithm analysis

As previously noted, the proposed BR decoding approach has two features: EM-like and consensus-like. Both can be explained as follows. The obvious similarity between the proposed decoding method and the EM algorithm lies in: (1) Both optimize the objective criterion by means of optimizing the corresponding auxiliary function; (2) Optimization is accomplished by iteration.

From the iterative procedure in Algorithm 3, we can define an auxiliary function of the objective function Eq. (1) as follows:

$$\mathcal{Q}(R; R', \mathcal{T}) = \sum_{k=1}^{|R'|} \sum_{w(a)} \gamma_{R'}(k, w(a)) \mathcal{L}(r_k, w(a)), \tag{7}$$

where R' is a previous hypothesis reference, R is the one to be generated by update, and loss criterion $\mathcal{L}(\cdot, \cdot)$ is as defined in Eq. (4). According to $\mathcal{L}(\cdot, \cdot)$ definition, Eq. (7) can be rewritten as

$$\mathcal{Q}(R; R', \mathcal{T}) = \sum_{k=1}^{|R'|} \sum_{w(a)} \gamma_{R'}(k, w(a))(1 - \delta(r_k, w(a))). \tag{8}$$

Since $\sum_{w(a)} \gamma_{R'}(k, w(a)) = 1$ for each k ($1 \leq k \leq |R'|$), we have

$$\mathcal{Q}(R; R', \mathcal{T}) = \sum_{k=1}^{|R'|} (1 - \gamma_{R'}(k, r_k)). \tag{9}$$

To minimize Eq. (9) for each position k ($1 \leq k \leq |R'|$), we thus select $r_k = \max_{w(a)} \gamma_{R'}(k, w(a))$. Similarly, with the help of auxiliary function Eq. (7), we can prove the convergence property of the proposed method advocated by the proof philosophy of EM convergence. We skip it over for simplicity. Note that the problem in this work is a discrete optimization problem, which makes the algorithm converge rapidly.

The consensus-like characteristics with the iterative BR decoding method can be observed from the $\gamma(\cdot, \cdot)$ accumulation procedure in Algorithm 2. Since each sentence in lattice $\mathcal{T}$ has a word arc a aligned to position k ($1 \leq k \leq |R|$) of the hypothesis reference R, we accumulate $\gamma(k, w) = \sum_{w(a)=w} \gamma(k, w)$ according to the word label to generate a competitive set, and the proposed method achieves the same goal as consensus decoding.

3 System combination

System combination in speech recognition is a promising way to improve the performance of the single recognizer. In general, it can produce up to 5% relative or more WER reduction taking the best individual recognizer as the baseline (Fiscus, 1997; Schwenk and Gauvain, 2000). Moreover, the output of the system combination can be further exploited to boost the recognition performance (Xu *et al.*, 2009b).

3.1 Previous approaches to system combination

The general principle of the popular system combination methods, such as ROVER, CNC, and time-frame error based system combination (Hoffmeister *et al.*, 2006) methods, can be for-

mulated as follows:

$$R^* = \text{argmin}_R \left\{ \sum_{\mathcal{W}} P(\mathcal{W}|\mathcal{O};\lambda^{(1)},\lambda^{(2)},...,\lambda^{(N)})\mathcal{L}(\mathcal{W},R) \right\}, \quad (10)$$

where λ refers to the parameter set of an individual recognizer and N is the component number.

Since it is not straightforward to obtain a solution to Eq. (10), approximation approaches have to be used, of which the ROVER and CNC methods are most popular, and the decision rule of the latter for each bin i ($1 \le i \le |R|$) is

$$w_i^* = \text{argmax}_{w_i} \sum_{i=1}^{N} \alpha_i P(w_i|\mathcal{O};\lambda^{(i)}), \quad (11)$$

where α_i ($0 \le \alpha_i \le 1$) is the corresponding component prior probability with the constraint $\sum_{i=1}^{N} \alpha_i = 1$.

3.2 Iterative Bayes risk approach to system combination

With the proposed scheme, we address the system combination problem with the following rule:

$$R^* = \text{argmin}_R \sum_{i=1}^{N} \alpha_i \sum_{\mathcal{W}} P_i(\mathcal{W}|\mathcal{O})\mathcal{L}(\mathcal{W},R), \quad (12)$$

where α_i has the same meaning as the one in Eq. (11). To do system combination with the proposed iterative BR method, the prominent feature is that no forced alignment operation is needed between individual systems. We interpolate only statistics $\gamma(k,w(a))$ with a linear method, so all the features for single lattice rescoring can be applicable to multiple system combination. The interpolation formula is

$$\gamma(k,w(a)) = \sum_{i=1}^{N} \alpha_i \gamma_i(k,w(a)). \quad (13)$$

Eq. (13) shows that single lattice decoding is actually a special case of the system combination with the proposed BR decoding method.

4 Experimental setup and evaluation

We report the experimental setup and the performance evaluation for the iterative BR decoding on both lattice rescoring and system combination in this section. By contrast, we also report the experimental results of CN in the case of lattice rescoring, and the results of ROVER and CNC in system combination.

4.1 Experimental setup

To achieve a comprehensive evaluation of the performance of the proposed algorithm for lattice rescoring and system combination, we trained six Mandarin acoustic systems. The overall 144.5 h training data is composed of two separate training sets: one is from the MSRA corpus that has 31.5 h data, and the other part with 113 h data is from China's 863 Project. Test data is 1.52 h long (1000 utterances in total), clean and provided by MSRA, Beijing.

Wave data was parameterized as mel frequency cepstral coefficients (MFCC) and perceptual linear prediction coefficients (PLP) and with energy, plus Δ and $\Delta\Delta$ features yielding a standard 39-dimensional feature vectors, respectively. All systems were trained with the HTK (Young *et al.*, 2008). We first trained two MFCC and PLP tri-phone cross-word maximum likelihood estimation (MLE) systems separately with 6.2k shared states and 16 Gaussians per state on the overall data. Then to have a variety of systems to do system combination, we conducted various discriminative approaches to training different systems. As a result, one pair of systems were trained with MPE while the other pairs with the boosted maximum mutual information (BMMI) (Povey *et al.*, 2008) using MFCC and PLP features, respectively. Language models were trigram and were also trained with the HTK language modeling tools over a 60k word dictionary, where text data was collected from various Chinese Web sites.

Word lattices were generated with language model scaling factor $\kappa_l = 12.0$ (refer to Eq. (5)). These word lattices were expanded to character lattices for testing. This is because a Chinese word is not an appropriate unit on which a performance metric can be based. We use the character error rate (CER) as a metric instead. Hence, the objective is to do BR decoding in terms of the minimum character error for the CN and the proposed decoding methods. For integrity, the specific statistic information of the character lattices is presented in Table 3.

The CER in Table 3 refers to the result obtained from the MAP decoding method, while the GER refers to the graph error rate, also known as the lattice oracle error rate, which is the lowest error rate for all the paths in the lattice. WGD_1 refers to

Table 3 Statistics for the test lattice with mel frequency cepstral coefficients (MFCC) as the front-end feature

System	CER (%)	GER (%)	WGD_1	WGD_2
MLE	17.95	5.3	62.41	50.35
BMMI	17.36	5.1	57.09	47.26
MPE	17.15	5.1	59.60	48.40

CER: character error rate; GER: graph error rate; WGD_1: word graph density; WGD_2: word graph depth

word graph density, i.e., word arcs in the lattice versus actual spoken words (Ortmanns and Ney, 1997); WGD_2 refers to word graph depth, i.e., the word arc number per second in the lattice (Povey, 2004)—both statistics actually describe the arc density of the lattice in different ways. Table 3 shows that models trained with discriminative methods typically generate obviously 'thinner' lattices but with better GER and CER.

To do lattice based BR decoding, we conducted two more optimized tasks on the character lattice. One is to prune the lattice, leading to the resulting lattice with 5.34 and 21.2 in terms of word graph depth for the CN and the iterative BR decoding methods, respectively. The other is to optimize lattice scaling factor κ for the two methods separately (Fig. 1).

4.2 Lattice scaling factor optimization

Since the lattice scaling factor can have an effect on the performances for the CN and the iterative BR decoding methods, it is worthwhile to investigate the performance variation versus the lattice scaling factor. Fig. 1 reports our experimental results on the MLE system with MFCC as the front-end feature.

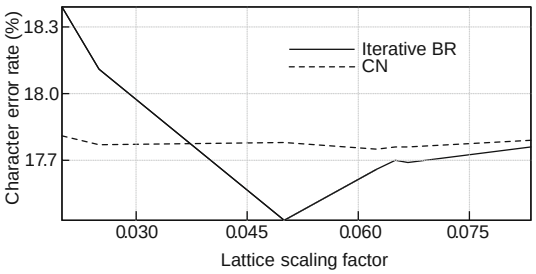

Fig. 1 Comparison between confusion network (CN) and iterative Bayes risk (BR) decoding methods of the character error rate (CER) versus the lattice scaling factor with mel frequency cepstral coefficients (MFCC) as the front-end feature

From Fig. 1, we observe that the two methods showed quite different characteristics in performance according to lattice scaling factor κ. CN was insensitive to lattice scaling and only about 0.1% variation was observed when κ was changed from 0.020 to 0.083 ($\kappa = 1/\kappa_l$). In contrast, the proposed iterative BR decoding method was much more sensitive to lattice scaling. It gained the best performance when the scaling factor was 0.05 in this case, and the gain was much degraded by further decreasing the scaling factor. Other experimental results revealed similar results as in Fig. 1. To make a better comparison, we chose $\kappa = 0.0625$ for both CN and the proposed methods in the following evaluations.

4.3 Convergence property illustration

As mentioned, the iterative BR decoding method is analogous to the EM scheme and each time the hypothesis reference R is updated, the objective function is decreased. Fig. 2 illustrates this property for both single lattice rescoring and system combination.

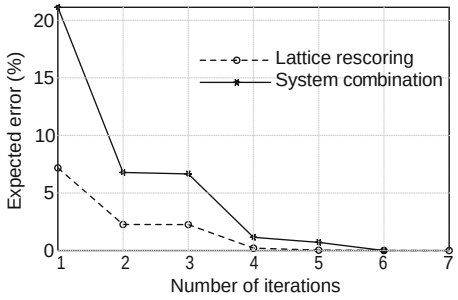

Fig. 2 Expected error versus the iteration number with mel frequency cepstral coefficients (MFCC) as the front-end feature. The number for single lattice rescoring is illustrated from the MPE trained system, while for system combination it is illustrated by the three-way system combination case

The vertical axis E in Fig. 2 was calculated as $E = \frac{1}{NC}\sum_{c=1}^{C}\sum_{n=1}^{N_i} E_n^{(c)}$, where C refers to the component number, N is the number of the entire test utterances, and N_i is the utterance number in the ith iteration. Fig. 2 indicates a fast convergence speed of the iterative BR decoding method. Estimates changed little from two to three iterations; this implies that the procedure would terminate with only three iterations in most cases, which was also illustrated by the much smaller E (approaching zero) after three iterations.

4.4 Lattice rescoring evaluation

Table 4 reports the results obtained with iterative BR decoding on single lattice rescoring where recognizers were trained with the MLE, BMMI, and MPE methods, respectively, taking PLP as the front-end feature.

Table 4 Single lattice rescoring with perceptual linear prediction coefficients (PLP) as the front-end feature

Decoding method	CER (%)		
	MLE	BMMI	MPE
Maximum a posteriori (MAP)	18.39	17.71	17.70
Confusion network (CN)	18.10	17.47	17.39
Iterative Bayes risk	17.92	17.28	17.37

CER: character error rate

From Table 4, the iterative BR method compared favorably with the CN method. Taking the MAP result as the baseline, it gained 2.28% reduced CER, while the CN method gained 1.56% reduced CER on average.

With a similar setup, but MFCC as the acoustic feature, further evaluation on the iterative BR decoding method for single lattice rescoring was obtained, as shown in Table 5.

Table 5 Single lattice rescoring with mel frequency cepstral coefficients (MFCC) as the front-end feature

Decoding method	CER (%)		
	MLE	BMMI	MPE
Maximum a posteriori (MAP)	17.95	17.36	17.15
Confusion network (CN)	17.68	17.09	16.62
Iterative Bayes risk	17.57	16.98	16.59

CER: character error rate

The results shown in Table 5 are similar to those in Table 4. In contrast with the MAP baseline, the iterative BR decoding method obtained 2.52% reduced CER, and the CN method obtained 2.05% CER reduction on average.

4.5 System combination evaluation

We evaluated the iterative BR decoding method for system combination. As a contrast, the performances of the ROVER and the CNC methods were also reported. In all cases, the prior $\alpha_i = 1/N$ was not optimized. For ROVER, the decision score was the mixture of word frequency and posterior probability, which was balanced with 0.6 and 0.4 accordingly, and no particular optimization was performed. To indicate the efficacy of the system combination decoding method, we picked the best MAP result as the baseline. Table 6 reports the experimental results for system combination with two- to six-component systems.

Table 6 System combination with different numbers of components*

Component number	CER (%)		
	ROVER	CNC	Iterative BR
2	17.09	15.79	15.70
3	16.33	15.71	15.59
4	16.36	15.66	15.56
5	16.92	15.70	15.61
6	16.74	15.84	15.79

* The baseline is 17.15. CER: character error rate

Table 6 shows that the proposed method gained the best performance in all cases, while ROVER gained the smallest accuracy improvements. Note that ROVER did not work well when two-component recognizers were combined despite the fact that the word posterior probability was taken into account. Also note that the other two combination methods gained the most accuracy improvement with the top two best system combinations, and that the CNC method gained 1.36% absolute accuracy improvement while the proposed iterative BR method gained 1.45% absolute improvement. Furthermore, compared with the baseline, the proposed method gained 9.27% relative accuracy improvement in its best setup, i.e., in the case of four-component system combination. Finally, Table 6 revealed that the performance of the system combination is not determined by the component number, but by the complementarity between the component systems.

5 Conclusions and future work

We have introduced an iterative Bayes risk decoding approach for lattice rescoring and system combination. It has similar functionality to the confusion network (CN), but is implemented with an entirely different framework. We approach the Bayes risk criterion as a discrete EM-like optimization problem, and the final result can be yielded with typically two to three iterations. Experiments showed that the proposed approach obtained better results than CN and confusion network combination

(CNC) in both lattice rescoring and system combination.

In future, we will investigate a framework to combine the proposed approach with ROVER and CNC to achieve better speech recognition.

Acknowledgements

Many thanks should be given to the two anonymous reviewers for their insightful suggestions; meanwhile, we would like to thank Daniel POVEY with Microsoft, Redmond, USA for his kind guidance on Haihua's research work.

References

Evermann, G., Woodland, P.C., 2000. Posterior Probability Decoding, Confidence Estimation, and System Combination. Proc. NIST Speech Transcription Workshop, College Park, MD.

Fiscus, J.G., 1997. A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). Proc. IEEE Workshop Automatic Speech Recognition and Understanding, p.347-354. [doi:10.1109/ASRU.1997.659110]

Goel, V., Kumar, S., Byrne, W.J., 2000. Minimum Bayes-risk automatic speech recognition. *Comput. Speech Lang.*, **14**(2):115-135. [doi:10.1006/csla.2000.0138]

Goel, V., Kumar, S., Byrne, W., 2004. Segmental minimum Bayes-risk decoding for automatic speech recognition. *IEEE Trans. Speech Audio Process.*, **12**(3):234-249. [doi:10.1109/TSA.2004.825678]

HakKani-Tur, D., Riccardi, G., 2003. A General Algorithm for Word Graph Matrix Decomposition. IEEE Int. Conf. on Acoustic, Speech, and Signal Processing, **1**:I-596-I-599. [doi:10.1109/ICASSP.2003.1198851]

Heigold, G., Macherey, W., Schluter, R., Ney, H., 2005. Minimum Exact Word Error Training. Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, p.186-190. [doi:10.1109/ASRU.2005.1566538]

Hoffmeister, B., Klein, T., Schluter, R., Ney, H., 2006. Frame Based System Combination and a Comparison with Weighted Rover and CNC. Int. Conf. on Spoken Language Processing, p.1-4.

Levenshtein, V.I., 1966. Binary codes capable of correcting deleltions, insertions and reversals. *Sov. Phys. Dokl.*, **10**:707-710.

Mangu, L., Brill, E., Stolcke, A., 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Comput. Speech Lang.*, **14**(4):373-400. [doi:10.1006/csla.2000.0152]

Ortmanns, S., Ney, H., 1997. A word graph algorithm for large vocabulary continuous speech recognition. *Comput. Speech Lang.*, **11**(1):43-72. [doi:10.1006/csla.1996.0022]

Povey, D., 2004. Discriminative Training for Large Vocabulary Speech Recognition. PhD Thesis, Cambridge University.

Povey, D., Woodland, P.C., 2002. Minimum Phone Error and I-Smoothing for Improved Discriminative Training. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, p.105-108. [doi:10.1109/ICASSP.2002.1005687]

Povey, D., Kanvevsky, D., Kingsbury, B., 2008. Boosted MMI for Model and Feature-Space Discriminative Training Recognition. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, p.4057-4060. [doi:10.1109/ICASSP.2008.4518545]

Schwenk, H., Gauvain, J.L., 2000. Combining Multiple Speech Recognizers Using Voting and Language Model Information. Int. Conf. on Spoken Language Processing, **2**:915-918.

Stolcke, A., Konig, Y., Weintraub, M., 1997. Explicit Word Error Minimization in N-Best List Rescoring. Proc. 5th European Conf. on Speech Communication and Technology, **1**:163-166.

Wessel, F., Schluter, R., Ney, H., 2001. Explicit Word Error Minimization Using Word Hypothesis Posterior Probabilities. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, **1**:33-36. [doi:10.1109/ICASSP.2001.940760]

Xu, H., Povey, D., Zhu, J., Wu, G., 2009a. Minimum Hypothesis Phone Error as a Decoding Method for Speech Recognition. INTERSPEECH, 10th Annual Conf. Int. Speech Communication Association, p.76-79.

Xu, H., Zhu, J., Wu, G., 2009b. An Efficient Multistage Rover Method for Automatic Speech Recognition. IEEE Int. Conf. on Multimedia and Expo, p.894-897. [doi:10.1109/ICME.2009.5202639]

Xu, H., Povey, D., Mangu, L., Zhu, J., 2010. An Improved Consensus-Like Method for Minimum Bayes Risk Decoding and Lattice Combination. IEEE Int. Conf. on Acoustics Speech and Signal Processing, p.4938-4941. [doi:10.1109/ICASSP.2010.5495100]

Young, S., Evermann, G., Gales, M., *et al.*, 2008. The HTK Book. Version 3.4, Cambridge University. Available from http://htk.eng.cam.ac.uk/