Evolutionary Minimum Verification Error Learning of the Alternative Hypothesis Model for LLR-based Speaker Verification

Yi-Hsiang Chao[1,2], Wei-Ho Tsai[3], Shih-Sian Cheng[1,2], Hsin-Min Wang[1], and Ruei-Chuan Chang[1,2]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2] Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
[3] Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan
{yschao,sscheng,whm}@iis.sinica.edu.tw, whtsai@en.ntut.edu.tw, rc@cc.nctu.edu.tw

Abstract

It is usually difficult to characterize the alternative hypothesis precisely in a log-likelihood ratio (LLR)-based speaker verification system. In a previous work, we proposed using a weighted arithmetic combination (WAC) or a weighted geometric combination (WGC) of the likelihoods of the background models instead of heuristic combinations, such as the arithmetic mean and the geometric mean, to better characterize the alternative hypothesis. In this paper, we further propose learning the parameters associated with WAC or WGC via an evolutionary minimum verification error (MVE) training method, such that both the false acceptance probability and the false rejection probability can be minimized. Our experiment results show that the proposed methods outperform conventional LLR-based approaches.

Index Terms: genetic algorithm, log-likelihood ratio, minimum verification error training, speaker verification

1. Introduction

The log-likelihood ratio (LLR) measure [1] is used in many speaker verification systems. Given an input utterance, U, the hypothesis test based on the LLR measure to determine whether or not U is spoken by the hypothesized speaker is expressed as

$$L(U) = \log p(U \mid \lambda) - \log p(U \mid \overline{\lambda}) - \theta \begin{cases} \geq 0 & \text{accept } H_0 \\ < 0 & \text{accept } H_1 \end{cases}, \quad (1)$$

where H_0 (the *null hypothesis*) represents that U is spoken by the hypothesized speaker; H_1 (the *alternative hypothesis*) represents that U is not spoken by the hypothesized speaker; θ is a decision threshold; λ is the hypothesized speaker model; and $\overline{\lambda}$ is the so-called anti-model or alternative hypothesis model. The $\overline{\lambda}$ model is usually ill-defined because, ideally, it should cover the space of all possible impostors. Many approaches have thus been proposed to characterize the $\overline{\lambda}$ model. One simple approach pools the speech data from a large number of background speakers, and trains a single speaker-independent model λ_0, called the world model or the Universal Background Model (UBM) [2]. The LLR measure in this case then becomes

$$L_1(U) = \log p(U \mid \lambda) - \log p(U \mid \lambda_0) - \theta . \quad (2)$$

Instead of using a single model to simulate potential impostors, a set of background models $\{\lambda_1, \lambda_2, \ldots, \lambda_B\}$ can be trained using speech from several representative speakers, called a cohort [3]. This leads to the following possible LLR measures, where the alternative hypothesis can be characterized by:

(i) the likelihood of the most competitive cohort model [4], i.e.,

$$L_2(U) = \log p(U \mid \lambda) - \max_{1 \leq i \leq B} \log p(U \mid \lambda_i) - \theta ; \quad (3)$$

(ii) the arithmetic mean of the likelihoods of the B cohort models [1], i.e.,

$$L_3(U) = \log p(U \mid \lambda) - \log\left\{ \frac{1}{B} \sum_{i=1}^{B} p(U \mid \lambda_i) \right\} - \theta ; \quad (4)$$

(iii) the geometric mean of the likelihoods of the B cohort models [4], i.e.,

$$L_4(U) = \log p(U \mid \lambda) - \frac{1}{B} \left\{ \sum_{i=1}^{B} \log p(U \mid \lambda_i) \right\} - \theta . \quad (5)$$

Obviously, the LLR measures $L_2(U)$, $L_3(U)$, and $L_4(U)$ are derived by heuristic combination methods that do not include an optimization process. Thus, the resulting system is far from optimal in terms of verification accuracy.

A more effective and robust LLR measure can be obtained by characterizing the alternative hypothesis as a weighted arithmetic combination (WAC) or a weighted geometric combination (WGC) of the likelihoods of the background models, instead of the above heuristic combinations [5]. The new combination scheme treats the background models unequally according to how close each individual is to the hypothesized speaker model, and quantifies the unequal nature of the background models by a set of weights optimized in the training phase. The optimization is performed by the minimum verification error (MVE) training method [5], which minimizes both the false acceptance probability and the false rejection probability.

Traditionally, MVE training has been realized by the gradient descent algorithm [5-7]; however, the algorithm only guarantees to converge to a local optimum. In this paper, we propose a new evolutionary MVE training method for learning the weights of the WAC- or WGC-based LLR measure. We embed the MVE training in a genetic algorithm (GA) [8], which is a widely used optimization algorithm that usually converges to a near global optimum. To do this, we incorporate a new mutation operator, called the one-step gradient descent operator (GDO), into the genetic algorithm.

The remainder of the paper is organized as follows. Section 2 presents the WAC- and WGC-based LLR measures and their relations to conventional LLR measures. In Section 3, we describe how we embed MVE training in the genetic algorithm. Section 4 details the experiment results. Finally, in Section 5, we present our conclusions.

2. WAC- and WGC-based LLR measures

We briefly review the WAC- and WGC-based LLR measures in this section. In the WAC-based LLR measure, the

likelihood of the alternative hypothesis model $\bar{\lambda}$, $p(U|\bar{\lambda})$, is expressed as

$$p(U|\bar{\lambda}) = \sum_{i=0}^{B} w_i p(U|\lambda_i), \quad (6)$$

while in the WGC-based LLR measure, $p(U|\bar{\lambda})$ is expressed as

$$p(U|\bar{\lambda}) = \prod_{i=0}^{B} p(U|\lambda_i)^{w_i}. \quad (7)$$

To ensure that the weights w_i, $i = 0,1,\ldots,B$, in Eqs. (6) and (7) satisfy $\sum_{i=0}^{B} w_i = 1$ and $w_i \geq 0, \forall i$, we represent w_i as a function of the intermediate parameters α_i, $i = 0,1,\ldots,B$, as follows [7],

$$w_i = \exp(\alpha_i) / \sum_{j=0}^{B} \exp(\alpha_j), \quad (8)$$

and solve α_i instead of w_i. Both WAC and WGC characterize the alternative hypothesis model $\bar{\lambda}$ by incorporating information available from the world model, λ_0, and the B cohort models, $\lambda_i, i = 1,...,B$. It is clear that WAC and WGC are equivalent to the arithmetic mean in $L_3(U)$ and the geometric mean in $L_4(U)$, respectively, when $w_0 = 0$ and $w_i = 1/B$, $i = 1,2,\ldots, B$, i.e., it is assumed that all the cohort models contribute equally. It is also clear that WAC and WGC will reduce to a maximum function in $L_2(U)$, if $w_{i^*} = 1$, $i^* = \text{argmax}_{1 \leq i \leq B} p(U|\lambda_i)$; and $w_i = 0, \forall i \neq i^*$. Furthermore, WAC and WGC will reduce to $L_1(U)$, if $w_0 = 1$ and $w_i = 0, \forall i \neq 0$. Thus, both WAC- and WGC-based LLR measures can be viewed as generalized and trainable versions of $L_1(U)$, $L_2(U)$, $L_3(U)$ or $L_4(U)$.

3. The genetic algorithm for MVE training

We propose a new evolutionary MVE training method that uses a genetic algorithm (GA) to train the weights w_i and the threshold θ in WAC- or WGC-based LLR measures.

Genetic algorithms (GA's) belong to a particular class of evolutionary algorithms (EA's) inspired by the process of natural evolution [8]. The operators involved in the evolutionary process are: encoding, parent selection, crossover, mutation, and survivor selection. GA's maintain a population of candidate solutions and perform parallel searches in the search space via the evolution of these candidate solutions.

To embed MVE training into a GA, the fitness function of the GA is represented as the overall expected loss function D of the MVE training method [5] calculated as,

$$D = x_0 \ell_0 + x_1 \ell_1, \quad (9)$$

where x_0 and x_1 reflect which type of error is of more concern than the other in a practical application; and ℓ_i is a loss function that describes the average false rejection errors ($i = 0$) or false acceptance errors ($i = 1$):

$$\ell_i = \frac{1}{N_i} \sum_{U \in H_i} s(d(U)), \quad (10)$$

where N_0 and N_1 are the numbers of utterances from the true speakers and the impostors, respectively; s is a sigmoid function $s(d(U)) = 1/[1+\exp(-a \cdot d(U))]$, where a is a scalar; and $d(U)$ is a mis-verification measure defined by

$$d(U) = \begin{cases} -L(U) & \text{if } U \in H_0 \\ L(U) & \text{if } U \in H_1 \end{cases}, \quad (11)$$

where $L(U)$ is the LLR measure defined in Eq. (1). By substituting WAC in Eq. (6) or WGC in Eq. (7) into Eq. (1), the goal of optimization is to find the weights w_i and the threshold θ that minimize the fitness function D in Eq. (9). This can be achieved by using a genetic algorithm.

Next, we describe the GA operators involved in our evolutionary MVE training method.

1) Encoding: Each chromosome is a string $\{\alpha_0, \alpha_1, ..., \alpha_B, \theta\}$ of length B+2, which is the concatenation of all intermediate parameters α_i in Eq. (8) and the threshold θ in Eq. (1). Chromosomes are initialized by randomly assigning a real value to each gene.

2) Parent selection: Five chromosomes are randomly selected from the population with replacement, and the one with the smallest fitness value is selected as a parent. The procedure is repeated iteratively until a pre-defined number (which is the same as the population size in this study) of parents is selected. This is known as *tournament selection* [8].

3) Crossover: The (B+1)-point crossover is used in this work. Two chromosomes are randomly selected from the parent population with replacement. The chromosomes can interchange each pair of their genes in the same positions according to a crossover probability *pc*.

4) Mutation: In most cases, the function of the mutation operator is to change the allele of the gene randomly in the chromosomes. For example, while mutating a gene of a chromosome, we can just draw a number from a uniform distribution at random, and add it to the allele of the gene. In this study, we use a new mutation operator, called the one-step gradient descent operator (GDO). The concept of the GDO is similar to that of the one-step K-means operator (KMO) [9, 10], which improves the fitness function after mutation by performing one iteration of the K-means algorithm. The GDO performs one iteration of gradient descent updating of the MVE training.

The one-step gradient descent operator (GDO) for the parameter α_i, $i = 0, 1, \ldots, B$, is defined as

$$\alpha_i^{new} = \alpha_i^{old} - \eta \frac{\partial D}{\partial \alpha_i}, \quad (12)$$

where α_i^{new} and α_i^{old} are, respectively, the parameter α_i in a chromosome after and before mutation; η is the step size; and $\frac{\partial D}{\partial \alpha_i}$ is computed by

$$\begin{aligned} \frac{\partial D}{\partial \alpha_i} &= x_0 \frac{\partial \ell_0}{\partial \alpha_i} + x_1 \frac{\partial \ell_1}{\partial \alpha_i} \\ &= x_0 \frac{\partial \ell_0}{\partial s} \cdot \frac{\partial s}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \alpha_i} + x_1 \frac{\partial \ell_1}{\partial s} \cdot \frac{\partial s}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \alpha_i} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U \in H_0} \left\{ a \cdot s(-L(U)) \left[1 - s(-L(U))\right] \cdot -\frac{\partial L}{\partial \alpha_i} \right\} \\ &\quad + x_1 \cdot \frac{1}{N_1} \sum_{U \in H_1} \left\{ a \cdot s(L(U)) \left[1 - s(L(U))\right] \cdot \frac{\partial L}{\partial \alpha_i} \right\}, \end{aligned} \quad (13)$$

where

$$\frac{\partial L}{\partial \alpha_i} = \sum_{j=0}^{B} \left(\frac{\partial L}{\partial w_j} \cdot \frac{\partial w_j}{\partial \alpha_i} \right) = w_i \left(\frac{\partial L}{\partial w_i} - \sum_{j=0}^{B} w_j \frac{\partial L}{\partial w_j} \right). \quad (14)$$

If WAC is used, then

$$\frac{\partial L}{\partial w_i} = \frac{-\partial}{\partial w_i} \log \left(\sum_{j=0}^{B} w_j p(U|\lambda_j) \right) = \frac{-p(U|\lambda_i)}{\sum_{j=0}^{B} w_j p(U|\lambda_j)}. \quad (15)$$

If WGC is used, then

$$\frac{\partial L}{\partial w_i} = \frac{-\partial}{\partial w_i}\left(\sum_{j=0}^{B} w_j \log p(U \mid \lambda_j) \right) = -\log p(U \mid \lambda_i). \quad (16)$$

The one-step gradient descent operator (GDO) for the threshold θ is defined as

$$\theta^{new} = \theta^{old} - \eta \frac{\partial D}{\partial \theta}, \quad (17)$$

where θ^{new} and θ^{old} are, respectively, the threshold θ in a chromosome after and before mutation; η is the step size; and $\frac{\partial D}{\partial \theta}$ is computed by

$$\begin{aligned} \frac{\partial D}{\partial \theta} &= x_0 \frac{\partial \ell_0}{\partial s} \cdot \frac{\partial s}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \theta} + x_1 \frac{\partial \ell_1}{\partial s} \cdot \frac{\partial s}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \theta} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U \in H_0} a \cdot s(-L(U))\left[1 - s(-L(U))\right] \\ &\quad - x_1 \cdot \frac{1}{N_1} \sum_{U \in H_1} a \cdot s(L(U))\left[1 - s(L(U))\right]. \end{aligned} \quad (18)$$

5) Survivor selection: We adopt the generational model [8], in which the whole population is replaced by its offspring.

4. Experiments

4.1. Experiment setup

We conducted speaker-verification experiments on speech data extracted from the XM2VTSDB multi-modal database [11]. In accordance with "Configuration II" described in [11], the database was divided into three subsets: "Training", "Evaluation", and "Test". We used "Training" to build each client model and the background models, and "Evaluation" to optimize the weights w_i in Eq. (6) or Eq. (7), along with the threshold θ. Then, the speaker verification performance was evaluated on "Test". As shown in Table 1, a total of 293 speakers[1] in the database were divided into 199 clients, 25 "evaluation impostors", and 69 "test impostors". Each speaker participated in 4 recording sessions at about one-month intervals, and each recording session consisted of 2 shots. In each shot, the speaker was prompted to utter 3 sentences "0 1 2 3 4 5 6 7 8 9", "5 0 6 9 2 8 1 3 7 4", and "Joe took father's green shoe bench out". Each utterance, sampled at 32 kHz, was converted into a stream of 24-order feature vectors, each consisting of 12 Mel-scale frequency cepstral coefficients [12] and their first time derivatives, by a 32-ms Hamming-windowed frame with 10-ms shifts.

We used all the clients' utterances from sessions 1 and 2 to train a world model (UBM), represented by a Gaussian mixture model (GMM) [1] with 512 mixture components. To implement $L_1(U)$, for each client, we used 12 (2×2×3) utterances/client from sessions 1 and 2 to generate the client model, represented by a GMM with 512 mixture components, through UBM-MAP adaptation [2]. To implement the other LLR measures, for each client, we used 12 (2×2×3) utterances/client from sessions 1 and 2 to generate the client model, represented by a GMM with 64 mixture components, by using the expectation-maximization (EM) algorithm [12]. For each client, the B closest speakers were chosen from the other 198 clients as the cohort [3] according to the degree of closeness measured in terms of the pairwise distance defined by [1]:

$$d(\lambda_i, \lambda_j) = \log \frac{p(U_i \mid \lambda_i)}{p(U_i \mid \lambda_j)} + \log \frac{p(U_j \mid \lambda_j)}{p(U_j \mid \lambda_i)}, \quad (19)$$

where λ_i and λ_j are speaker models trained using the i-th speaker's utterances U_i and the j-th speaker's utterances U_j, respectively. In the experiments, B was set to 20, and each cohort model was represented by a GMM with 64 mixture components

Table 1. *Configuration of the speech database.*

Session	Shot	199 clients	25 impostors	69 impostors
1	1	Training	Evaluation	Test
	2			
2	1			
	2			
3	1	Evaluation		
	2			
4	1	Test		
	2			

To optimize the weights, w_i, and the threshold, θ, we used 6 utterances/client from session 3, along with 24 (4×2×3) utterances/evaluation-impostor over the four sessions, which yielded 1,194 (6×199) client samples and 119,400 (24×25×199) impostor samples. To speed up the MVE training process, only 2,250 imposter samples randomly selected from 119,400 such samples were used. In the performance evaluation, we tested 6 utterances/client in session 4 and 24 utterances/test-impostor over the four sessions, which involved 1,194 (6×199) client trials and 329,544 (24×69×199) impostor trials. We use the Detection Error Tradeoff (DET) curve [13] for the performance evaluation. In addition, we use the Half Total Error Rate (HTER), which reflects the performance at a single operating point on the DET curve. The HTER is defined as

$$\text{HTER} = (P_{Miss} + P_{FalseAlarm}) / 2, \quad (20)$$

where P_{Miss} is the miss (false rejection) probability and $P_{FalseAlarm}$ is the false alarm (false acceptance) probability. It is clear that the loss functions ℓ_0 and ℓ_1 in Eq. (10) will approximate P_{Miss} and $P_{FalseAlarm}$, respectively, if the scalar a in the sigmoid function is set to a sufficiently large value. In our experiments, a was set to 10, and x_0 and x_1 in the fitness function D in Eq. (9) were set to 0.5. Thus, the minimization of the fitness function D in Eq. (9) is equivalent to the minimization of the HTER.

4.2. Experiment results

We employed the proposed evolutionary MVE training methods in two LLR measures: 1) WAC with the world model plus the 20 closest cohort models ("WAC_GA_w_20c"); and 2) WGC with the world model plus the 20 closest cohort models ("WGC_GA_w_20c"). The population size of the GA was set to 100, and the crossover probability pc was set to 0.5. We also implemented MVE training using the gradient descent algorithm in two LLR measures: 1) WAC with the world model plus the 20 closest cohort models ("WAC_GD_w_20c"); and 2) WGC with the world model plus the 20 closest cohort models ("WGC_GD_w_20c").

For the performance comparison, we used five systems as our baselines: 1) $L_1(U)$, using a 512-mixture client GMM

[1] We omitted 2 speakers (ID numbers 313 and 342) because of partial data corruption.

through UBM-MAP adaptation ("$L1$_MAP"); 2) $L_1(U)$, using a 64-mixture client GMM through EM training ("$L1$"); 3) $L_2(U)$ with the 20 closest cohort models ("$L2$_20c"); 4) $L_3(U)$ with the 20 closest cohort models ("$L3$_20c"); and 5) $L_4(U)$ with the 20 closest cohort models ("$L4$_20c").

Fig. 1 shows the DET curves evaluated on "Test" for the speaker verification performance achieved by various methods, and Table 2 summarizes the experiment results in terms of HTER. For each baseline, the value of the decision threshold θ was tuned to minimize HTER on "Evaluation", and then applied to "Test". The decision thresholds and the weights of the WAC- and WGC-based LLR measures were optimized automatically using "Evaluation", and then applied to "Test". From Fig. 1, we observe that both WAC- and WGC-based LLR measures significantly outperform all the baseline systems. Table 2 shows that "WAC_GA_w_20c" and "WGC_GA_w_20c" outperform "WAC_GD_w_20c" and "WGC_GD_w_20c", respectively. It is clear that each of the WAC and WGC methods achieved a relative improvement of more than 10% over the baseline systems. Incidentally, the baseline system "$L1$" outperformed "$L1$_MAP", a well-recognized state-of-the-art method for text-independent speaker verification, and the other baseline systems. This may be because the training and test utterances in the XM2VTSDB database have the same content.

5. Conclusions

We have proposed using more comprehensive LLR measures based on improved characterization of the alternative hypothesis model for speaker verification. The alternative hypothesis model is built on either a weighted arithmetic combination (WAC) or a weighted geometric combination (WGC) of useful information extracted from a set of pre-trained background models. The parameters associated with the WAC or WGC are optimized via a new evolutionary minimum verification error (MVE) training method, such that both the false acceptance probability and the false rejection probability are minimized. Our experiment results demonstrate that the proposed methods outperform conventional LLR-based approaches.

6. Acknowledgements

This work was funded in part by the National Science Council, Taiwan, under Grant NSC95-2221-E-001-034.

7. References

[1] Reynolds, D. A., "Speaker Identification and Verification Using Gaussian Mixture Speaker Models", Speech Communication, vol.17, pp. 91-108, 1995.

[2] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B., "Speaker Verification Using Adapted Gaussian Mixture Models", Digital Signal Processing, vol. 10, pp. 19-41, 2000.

[3] Rosenberg, A. E., Delong, J., Lee, C. H., Juang, B. H., and Soong, F. K., "The Use of Cohort Normalized Scores for Speaker Verification", in Proc. ICSLP1992.

[4] Liu, C. S., Wang, H. C., and Lee, C. H., "Speaker Verification Using Normalized Log-Likelihood Score", IEEE Trans. Speech and Audio Processing, vol. 4, pp. 56-60, 1996.

[5] Chao, Y. H., Tsai, W. H., Wang, H. M., and Chang, R. C., "Improved Methods for Characterizing the Alternative Hypothesis Using Minimum Verification Error Training for LLR-based Speaker Verification", in Proc. ICASSP2007.

[6] Rosenberg, A. E., Siohan, O., and Parthasarathy, S., "Speaker Verification Using Minimum Verification Error Training", in Proc. ICASSP1998.

[7] Chou, W. and Juang, B. H., Pattern Recognition in Speech and Language Processing, CRC Press, 2003.

[8] Eiben, A. E. and Smith, J. E., Introduction to Evolutionary Computing, Springer, Berlin, 2003.

[9] Krishna, K. and Narasimha Murty, M., "Genetic K-Means Algorithm", IEEE Trans. Systems, Man, and Cybernetics, vol. 29, no. 3, June 1999.

[10] Cheng, S. S., Chao, Y. H., Wang, H. M., and Fu, H. C., "A Prototypes Embedded Genetic Algorithm for K-means Clustering", in Proc. ICPR2006.

[11] Luettin, J. and Maître, G., Evaluation Protocol for the Extended M2VTS Database (XM2VTSDB), IDIAP-COM 98-05, IDIAP, 1998.

[12] Huang, X., Acero, A., and Hon, H. W., Spoken Language Processing, Prentics Hall, New Jersey, 2001.

[13] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M., "The DET Curve in Assessment of Detection Task Performance", in Proc. Eurospeech1997.

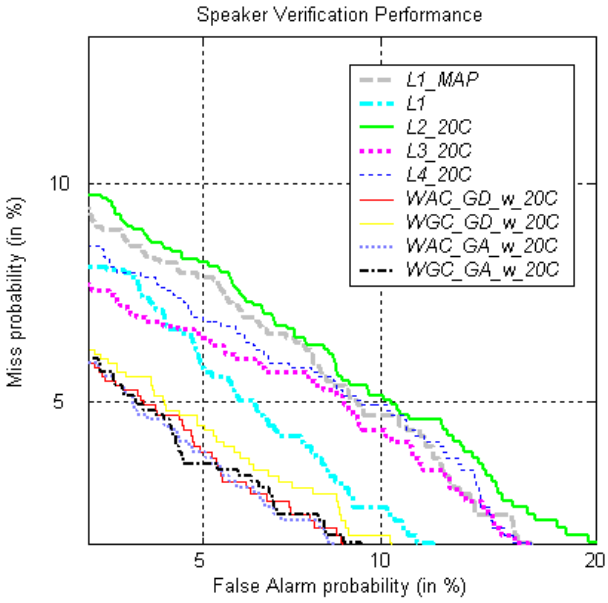

Figure 1: *DET curves for "Test".*

Table 2. *Experiment results in terms of HTER.*

Methods	min HTER for "Evaluation"	HTER for "Test"
$L1$_MAP	0.0714	0.0626
$L1$	0.0651	0.0545
$L2$_20c	0.0776	0.0635
$L3$_20c	0.0676	0.0535
$L4$_20c	0.0734	0.0583
WAC_GD_w_20c	0.0535	0.0448
WGC_GD_w_20c	0.0522	0.0453
WAC_GA_w_20c	0.0516	0.0443
WGC_GA_w_20c	0.0489	0.0437