Adaptive Gaussian Backend for Robust Language Identification

Mitchell McLaren, Aaron Lawson, Yun Lei, Nicolas Scheffer

Speech Technology and Research Laboratory, SRI International, California, USA

{mitch,aaron,yunlei,scheffer}@speech.sri.com

Abstract

This paper proposes adaptive Gaussian backend (AGB), a novel approach to robust language identification (LID). In this approach, a given test sample is compared to language-specific training data in order to dynamically select data for a trial-specific language model. Discriminative AGB additionally weights the training data to maximize discrimination against the test segment. Evaluated on heavily degraded speech data, discriminative AGB provides relative improvements of up to 45% and 38% in equal error rates (EER) over the widely adopted Gaussian backend (GB) and neural network (NN) approaches to LID, respectively. Discriminative AGB also significantly outperforms those techniques at shorter test durations, while demonstrating robustness to limited training resources and to mismatch between training and testing speech duration. The efficacy of AGB is validated on clean speech data from National Institute of Standards and Technology (NIST) language recognition evaluation (LRE) 2009, on which it was found to provide improvements over the GB and NN approaches.

Index Terms: language recognition, adaptive Gaussian backend, support vector machines, noisy speech

1. Introduction

Since the recent introduction of iVectors, the Gaussian backend (GB) approach has become mainstream for language identification (LID). Other, more complex techniques have been evaluated [1], with GB remaining the simplest and state-of-the-art solution to LID using iVectors. In the context of heavily degraded speech data, however, iVector distributions are often far from Gaussian (as shown later in this article) limiting the applicability of GB to such cases.

First proposed in the field of speaker identification (SID), iVectors have benefited from considerable improvements due to the tailoring of classification techniques. In contrast to LID, the task of SID is determining whether two speech samples (or iVectors) originate from the same speaker. In this paper, we leverage this same speaker-comparison paradigm to find a subset of training data for each trial. Specifically, a test iVector is directly compared to iVectors of a known language from the training set to find the most relevant training data for that test. We demonstrate that adapting the mean of the traditional GB scoring approach with respect to the trial-specific training data provides significant improvements for heavily degraded speech data from the DARPA Robust Automatic Transcription of Speech (RATS) program. These improvements are also shown to generalize to the clean speech data of the NIST 2009 LRE.

This paper proposes the use of adaptive Gaussian backend (AGB) for LID with iVectors. This approach defines a trial-specific language model mean based on the top-N scores when comparing the test to individual training segments. Support vector machines (SVMs) are used to improve performance through discriminatively selecting and weighting the training samples used for the mean. The robustness of the approach to limited training data is highlighted along with a short analysis on computational cost and trials on clean speech data.

The structure of the paper is as follows: Section 2 presents the Gaussian backend and neural network approaches to LID. Section 3 proposes the novel adaptive Gaussian backend approaches. Section 4 details the experimental protocol and Section 5 presents the results and analysis.

2. State-of-the-Art Language Identification

Recent advances in language identification have often leveraged work from the field of speaker identification, particularly using iVectors as features. Prior to the iVector paradigm, LID research tended to follow two research tracks: phone token sequence based modeling (PPRLM) and acoustic feature-based modeling. As with SID, LID modeling with acoustic features progressed from GMM-UBM frame-scoring techniques to averaged-frame scoring with support vector machines [2], evolving to GMM-supervector scoring [3]. Again paralleling research in SID, various subspace modeling techniques have been applied: JFA/LFA in the GMM space [4], and nuisance attribute projection in the SVM modeling approach [5]. Both techniques have been applied to the feature space for LID as well (e.g., fNAP [6], for example), though often with limited success.

In SID, the iVector-PLDA framework greatly reduced the complexity of previous approaches while improving accuracy [7]. Due to the multi-class nature of the LID problem, the Gaussian backend and neural network approaches have emerged as the state-of-the-art in recent years.

2.1. Gaussian Backend

The Gaussian backend [8] has served as the scoring method for many basic approaches to LID in which data is assumed to be normally distributed. In contrast to the traditional use of supervectors or pure acoustic features, the compactness and high density of information contained in the iVector representation lends itself well to relatively straightforward scoring approaches. Consequently, scoring techniques such as GB are often optimal for LID tasks in the iVector paradigm, where the data satisfies Gaussian assumptions and little gain on clean speech is achieved with more complex scoring [1].

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract D10PC20024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA or its contracting agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch. "A" (Approved for Public Release, Distribution Unlimited)

The GB likelihood for LID can be formulated as $s_{lt} = \boldsymbol{w}_t \boldsymbol{W}^{-1} \boldsymbol{\mu}_l - \frac{1}{2} \boldsymbol{\mu}_l \boldsymbol{W}^{-1} \boldsymbol{\mu}_l$ where $\boldsymbol{W}$ is the within-class covariance of training iVectors, $\boldsymbol{\mu}_l$ the mean of the training iVectors from language l and $\boldsymbol{w}_t$ the test iVector. Normalizing all i-vectors for $\boldsymbol{W}$ simplifies this to,

$$s_{lt} = \boldsymbol{w}_t \boldsymbol{\mu}_l - \frac{1}{2} \boldsymbol{\mu}_l \boldsymbol{\mu}_l. \quad (1)$$

2.2. Neural Networks

A neural network (NN) can be utilized as another successful backend for language identification under the iVector framework. Particularly in the case of heavily degraded RATS data, the robustness of NN has been demonstrated [9]. In contrast to the GB, NN requires a more tedious training strategy as NN is renowned for being sensitive to characteristics of the training data and over-fitting. Consequently, considerable research has resulted in numerous training regimes for the NN. In this work a three-layer feed-forward NN is trained directly on iVectors using the mini-batch gradient descent algorithm with cross-entropy error back-propagation [10]. iVectors are normalized by the mean and variance or the training set. A sigmoid function for nonlinearity is used for both hidden and output layers and weights are updated with momentum to speed up the training and avoid local minima [10].

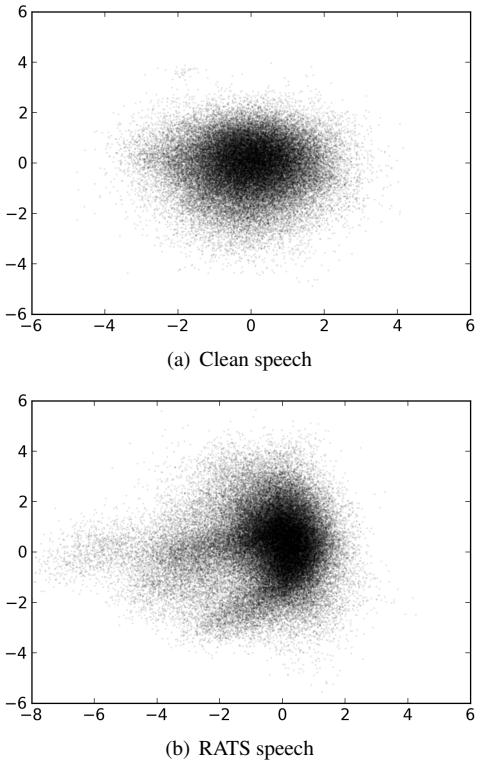
3. Adaptive Gaussian Backend (AGB)

Common in machine learning is the modeling of some training data from a certain class (e.g., speaker or language), and then at test time, a previously unseen sample is compared to the model to produce a score (likelihood or other). Models make assumptions about the data and in the case of GB, the model assumes the data has a Gaussian distribution. Figure 1 depicts the distribution of iVectors from (a) clean data from the LRE training data and (b) the RATS LID training data as projected into the first two dimensions of a principal component analysis (PCA) space learned on the same data (see Section 4 for dataset details). Figure 1 clearly shows that the distributions are different with the clean data being closer to the Gaussian assumptions of the iVector subspace model and GB than the RATS data. To quantify the level of "skewness" from a normal distribution of the iVectors from each language, we measured the kurtosis of to the 400 dimensions of iVectors on a per-language basis. The average of these across languages was found to be 0.32 for LRE iVectors and 0.52 for RATS iVectors. It is the non-Gaussian distribution of the RATS speech data that motivated the proposal of *adaptive* Gaussian backend (AGB).

In the proposed AGB approach, the traditional GB is adapted at trial time according to the characteristics of the given test segment. In essence, a subset of GB training data is dynamically selected and used to adapt the mean, $\boldsymbol{\mu}_l$, of the GB model to a test-specific mean, $\boldsymbol{\mu}_{lt}$. That is, Eq. (1) becomes,

$$s_{lt} = \boldsymbol{w}_t \boldsymbol{\mu}_{lt} - \frac{1}{2} \boldsymbol{\mu}_l \boldsymbol{\mu}_{lt}. \quad (2)$$

We found it necessary to retain global mean $\boldsymbol{\mu}_l$ in the bias term to maintain numeric stability as $\boldsymbol{\mu}_{lt}$ is typically estimated from relative few iVectors. We present two methods of estimating the adapted mean $\boldsymbol{\mu}_{lt}$: (1) Top-N averaging and (2) discriminative averaging. These techniques leverage advancements made in speaker identification with particular regard to simplistic dot scoring for segment comparison [7], and the inherent ability of support vector machines for data selection [11].



(a) Clean speech

(b) RATS speech

Figure 1: Illustrating the distribution of iVectors extracted from clean LRE data and heavily degraded RATS speech data.

3.1. Top-*N* AGB

The Euclidean norm is a common distance measure for the similarity of two vectors (often referred to as dot-scoring). This accurate and computationally efficient scoring technique is used after normalizing by the within-class covariance to compare the test segment against all training data from a given target language from which the top-N closest iVectors are used to estimate $\boldsymbol{\mu}_{lt}$. Specifically, the training dataset, $\boldsymbol{T}$, can be broken down into subsets $\boldsymbol{T}_l$ for each target language l. At trial time, the test iVector is compared to each $\boldsymbol{T}_l$, via the dot-product, and the mean $\boldsymbol{\mu}_{lt}$ for Eq. (2) is the average of the top-N closest iVectors $\boldsymbol{\mu}_{lt}$. In Section 5 we broadly illustrate the effect of N on LID performance with a preference for a small selection ($N = 30$).

3.2. Discriminative AGB

Extending the straightforward top-N selection, we utilize support vector machines (SVMs) to dynamically select and weight the training data that contain the most discriminative information. SVMs are a discriminative classifier which, when trained with two classes of training data labeled $y = \{-1, 1\}$, find a dividing hyperplane that maximizes the margin between the classes. The hyperplane is constructed from a small subset of weighted vectors from each class—these are termed *support vectors*. The normal to the trained hyperplane can be expressed as a combination of weighted iVectors $\boldsymbol{v} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{w}_{li}$, where $\boldsymbol{w}_i$ are the iVectors used to train the SVM with corresponding class labels, y_i, and learned coefficients or weights α_i. The SVM training phase can be viewed as a data-selection regime in which the most discriminative samples are selected

and weighted with $\alpha_i > 0$. Previous research has shown support vector selection using a linear kernel to be useful for dataset refinement [11] and the same regime is extended in this paper to the task of dynamic and discriminative data selection for LID.

In this work, the training vectors from a language represent one class ($y = -1$) and the test sample from unknown language represents the alternate class ($y = 1$). This differs from typical SVM use for LID in which the hyperplane is learned to discriminate languages rather than a single test example against vectors from a language. Of interest in this training regime is that the SVM attempts to discriminate the test segment against the training data even if they are from the same language.

Support vectors selected from the training data (class label $y = -1$) can be utilized in the Top-N scoring approach instead of selecting a fixed number of training iVectors using the Euclidean distance. Referred to as Top-SV in Section 3, the benefit of this approach is that N is not fixed and dynamically and discriminatively selected for each test and language pair.

In light of adapting the mean of the Gaussian backend in a discriminative manner, the coefficients can be utilized to find the weighted sum of the selected training iVectors. That is, the AGB mean for Eq. (2) becomes,

$$\boldsymbol{\mu}_{lt} = \sum_{i}^{N} \boldsymbol{w}_{li} \frac{\alpha_{ti}}{\sum \alpha_{ti}}. \tag{3}$$

where the iVectors $\boldsymbol{w}_{li}$ are the support vectors of class $y = -1$ when training an SVM to discriminate test $\boldsymbol{w}_t$ against iVectors from language l. This approach is referred to as discriminative AGB (D-AGB).

4. Experimental Protocol

The RATS LID task involves five target languages (Farsi, Urdu, Pashto, Arabic Levantine and Dari) for which conversational telephone recordings were retransmitted over seven channels for the RATS program (the eighth channel "D" was excluded from the LID task). The signal-to-noise ratio (SNR) of retransmitted signals ranged between 30dB to 0dB with very heavy degradation. The total amount of system training data was 76,552 segments distributed across the languages as follows: Farsi (3%), Urdu (15%), Pashto (23%), Arabic Levantine (32%), Dari (1%) and non-target (25%). The evaluation data consisted of 2,761 segments cut to durations of 3s, 10s, 30s, and 120s and was more evenly distributed: Farsi (18%), Urdu (15%), Pashto (14%), Arabic Levantine (15%), Dari (3%) and non-target (34%).

Mel-frequency cepstral coefficients (MFCC) features of 7 dimensions were extracted for each audio segment and appended with '7-1-3-7' shifted delta cepstrum [12], resulting in a final vector of 56 dimensions. In conjunction with a 2048-component diagonal covariance Universal Background Model (UBM), iVectors were extracted from a 400-dimensional subspace. The iVector subspace, Gaussian backend, neural network and proposed approaches were trained from the full training data set. LID performance was evaluated under the same regime as used in the RATS program by reporting the equal error rate (EER) after pooling scores of each test segment against each target language. Duration-dependent score calibration was performed using multi-class logistic regression following the strategy of [9]. Calibration was learned and applied through cross-validation of two sets for which care was taken to ensure sets were disjoint in terms of original recording.

Table 1: EER of proposed techniques against NN and GB approaches for different test durations on the RATS LID task.

	Test duration			
Scorer	**3s**	**10s**	**30s**	**120s**
GB	27.3%	19.1%	12.4%	7.9%
NN	28.1%	18.1%	10.1%	**4.2%**
Top-15	23.6%	13.4%	8.3%	4.7%
Top-30	22.7%	12.8%	8.0%	5.1%
Top-60	22.6%	12.7%	8.4%	5.6%
Top-SV	24.3%	13.7%	8.7%	6.2%
D-AGB	**22.4%**	**11.3%**	**6.8%**	4.4%

Experiments using the NIST Language Recognition Evaluation (LRE) 2009 data followed the "open-set" protocol of the official evaluation plan from NIST [13] with performance reported in terms of C_{avg}. The LRE data consisted of 23 target languages while system training data was sourced from an additional 28 languages as per [14]. The iVector extractor was trained from nearly 76,855 clean speech segments and the UBM a language-balanced subset of 20,726 segments thereof. The backends were trained using 58,210 segments of the training set found by discarding those with less that 30s of audio. This filtering brought a consistent improvement to all classifiers across the 3s, 10s and 30s test durations.

5. Results

The following section benchmarks the proposed AGB scoring techniques against the widely adopted Gaussian backend and neural network approaches. Robustness of each technique to training data availability is then analyzed. The applicability of AGB to clean speech data is demonstrated before providing a short analysis of computationally efficiency.

5.1. Adaptive Gaussian Backend

Performance of the proposed AGB techniques were evaluated on the RATS data with results presented in Table 1 alongside the GB and NN techniques as described in Section 2. Comparing traditional techniques, NN was found to outperform GB by 46% relative when 120 seconds of test speech was available. This reduced to 18% and 5% along with duration to 30 and 10 seconds, respectively. For three seconds of speech GB and NN were comparable. The preference for longer durations with NN can be attributed to the training data being of similar length.

The proposed Top-N AGB technique was optimal at $N = 30$, outperforming NN for durations less than 120 seconds and providing relative improvements of 17–36% over GB. Discriminative AGB (D-AGB) provided the best overall performance with more than 35% relative improvement over NN in the 10s and 30s cases, and a considerable 45% over GB in the 120s test case while providing the best performance on the 3s tests. These results illustrate not only that AGB is more robust to heavily degraded speech than traditional approaches, but also is robust to mismatch in duration between training and testing data. In light of being a largely tuneless system, the performance benefit of D-AGB over NN at the cost of computation (see Section 5.4) may be highly desirable.

For comparative purposes, the system Top-SV uses the SVM to dynamically select support vectors as the training vectors from which the mean is estimated for AGB. Comparing Top-SV and D-AGB illustrates the positive effect of weighting

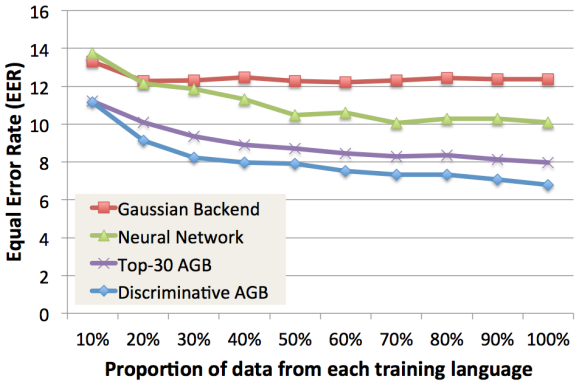


Figure 2: EER for the 30s RATS LID task using GB, NN, Top-30 AGB and D-AGB methods as a function of training data.

Table 2: $C_{\text{avg}} \times 100$ of proposed AGB techniques against NN and GB approaches on NIST LRE'09 data.

	Test duration		
Scorer	**3s**	**10s**	**30s**
GB	16.4	7.9	4.9
NN	19.8	10.1	5.9
Top-60	14.9	7.0	4.3
D-AGB	**14.4**	**6.5**	**3.6**

training data by the support vector coefficients rather than averaging iVectors as in the Top-N approach. This is particularly effective for longer durations with up to 40% relative improvement attributed to using the coefficient weights.

5.2. Scaling with Training Data

In this section, we analyze the robustness of the GB, NN and the proposed AGB techniques to data availability. Figure 2 depicts the EER of the 30s RATS LID task for each scoring technique as a function of the available training data. The number of hidden layers in the NN was set to 200 and N=30 for Top-N AGB (limited gains were found through tuning either parameter). The figure indicates that with only 20% of available data, GB reaches its full potential. In contrast, NN continues to improve as more training data becomes available. The proposed AGB approaches consistently outperform both GB and NN irrespective of data availability. Both techniques demonstrate considerable robustness to limited data, and similar to NN, continue to improve with more data. Top-30 AGB offered an absolute gain of 1.8–2.5% over NN across the range of data availability while the discriminative AGB offered an absolute gain of 2.6–3.7%. Additionally, the dynamic nature of AGB facilitates the use of new training data to be incorporated, which is a highly desired trait when in-the-field adaptability is required.

5.3. Clean Speech Experiments

The previous section compared the robustness of the proposed AGB techniques against the widely adopted GB and NN techniques in the context of heavily degraded speech. While developed for this scenario, this section provides insight into the applicability of AGB to clean speech data. Results from each technique when evaluated on the NIST LRE'09 dataset are presented in Table 2. In contrast to trends observed on RATS data, Table 2 shows traditional GB provided superior performance to the NN on clean speech. This is likely due to additional difficulty in learning 24 language classes from less data than the 6 classes learned in the RATS case. Similar to the RATS data, however, the proposed AGB approaches consistently outperformed the GB. The simpler Top-60 flavor of AGB provided 6-13% relative gains over GB ($N = 60$ provided a small and consistent gain over $N = 30$ on the LRE data) whereas D-AGB offered a 12-26% relative improvement over GB and was particularly effective on longer durations.

Table 3: Computation time in CPU seconds to evaluate the RATS 30s test iVectors using different scoring techniques.

CPU sec.	**GB**	**NN**	**Top-30**	**D-AGB**
Train	5.7	14727	5.7	5.7
Test	0.1	0.7	50	554

5.4. Computation

The Gaussian backend and neural network utilize a static model to provide a rapid scoring process. The dynamic AGB, on the other hand, can be expected to have a greater computational load. To quantify this load, Table 3 provides in CPU seconds the computation time to train GB, AGB and NN models and to evaluate each scoring technique on the 2761 test iVectors from the RATS 30s LID task across the six language classes. Although the NN takes a considerable time to train models (which can be done offline), it is very efficient for evaluation, taking less than a CPU second. Similarly, the GB is the most efficient of the classifiers. The proposed Top-30 AGB method demands 70 times more computation than NN for evaluation and D-AGB a magnitude more. An advantage of the proposed methods over traditional approaches (in addition to those advantages stated in previous sections) is that system tuning is minimal except in the case of Top-N where N is a tunable parameter. Dependent on the application, the robustness and accuracy benefits versus computational trade-off may be acceptable and desirable.

6. Conclusion

We presented a novel approach to language identification termed adaptive Gaussian backend (AGB), in which a subset of language-specific training data was dynamically selected and weighted to produce a trial-dependent language model mean. This training subset was selected at trial time using simple dot-scoring techniques or the discriminative optimization of SVMs. Evaluated on heavily degraded speech data from the RATS LID task, improvements of up to 45% and 38% were observed over state-of-the-art Gaussian backend and neural network approaches, respectively. AGB demonstrated robustness to training data availability, short test durations and duration mismatch between test and training data. This robustness was shown to come at a cost of trial-time computation. Finally, the proposed techniques were found to be improve on state-of-the-art techniques on clean speech data from NIST LRE'09.

7. References

[1] M. Penagarikano, A. Varona, M. Diez, L. J. Rodriguez-Fuentes, and G. Bordel, "Study of different backends in a state-of-the-art language recognition system," in *Proc. Interspeech*, 2012.

[2] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2, pp. 210–229, 2006.

[3] C. H. You, K. A. Lee, and H. Li, "An SVM kernel with GMM-supervector based on the Bhattacharyya distance for speaker recognition," *Signal Processing Letters, IEEE*, vol. 16, no. 1, pp. 49–52, 2009.

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[5] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. IEEE ICASSP*, vol. 1, 2006, pp. 97–100.

[6] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Compensation of nuisance factors for speaker and language recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 15, no. 7, pp. 1969–1978, 2007.

[7] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. Interspeech*, 2009, pp. 1559–1562.

[8] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Language score calibration using adapted Gaussian backend," in *Proc. Interspeech*, 2009, pp. 2191–2194.

[9] P. Matejka, O. Plchot, M. Soufifar, O. Glembek, L. F. DHaro, K. Veselỳ, F. Grézl, J. Ma, S. Matsoukas, and N. Dehak, "Patrol team language identification system for DARPA RATS P1 evaluation," in *Proc. Interspeech*, 2012.

[10] C. M. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006, vol. 4, no. 4.

[11] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, "Data-driven background dataset selection for SVM-based speaker verification," *IEEE Trans. on Speech and Audio Processing*, vol. 18, no. 6, pp. 1496–1506, 2010.

[12] B. Bielefeld, "Language identification using shifted delta cepstrum," in *Proc. Fourteenth Annual Speech Research Symposium*, 1994.

[13] National Institute of Standards and Technology, *The 2009 NIST Language Recognition Evaluation Plan*, available: http://www.itl.nist.gov/iad/mig/tests/lre/2009/.

[14] D. Martinez, L. Burget, L. Ferrer, and N. Scheffer, "iVector-based prosodic system for language identification," in *Proc. IEEE ICASSP*, 2012, pp. 4861–4864.