TOWARDS A MORE EFFICIENT SVM SUPERVECTOR SPEAKER VERIFICATION SYSTEM USING GAUSSIAN REDUCTION AND A TREE-STRUCTURED HASH

SAND2013-6633C

Richard D. McClanahan
Sandia National Laboratories
Albuquerque, New Mexico, USA
rmcclan@sandia.gov

Phillip L. De Leon
New Mexico State University
Las Cruces, New Mexico USA
pdeleon@nmsu.edu

ABSTRACT

Speaker verification (SV) systems that employ maximum a posteriori (MAP) adaptation of a Gaussian mixture model (GMM) universal background model (UBM) incur a significant test-stage computational load in the calculation of a posteriori probabilities and sufficient statistics. We propose a multi-layered hash system employing a tree-structured GMM which uses Runnalls' Gaussian mixture reduction technique. The proposed method is applied only to the test stage and does not require any modifications to the training stage or previously-trained speaker models. With the tree-structured hash system we are able to achieve a factor of $8\times$ reduction in test-stage computation with no degradation in accuracy. Furthermore, we can achieve computational reductions greater than $21\times$ with less than 7.5% relative degradation in accuracy.

1. INTRODUCTION

- The objective of speaker verification (SV) is to accept or reject a identity claim based on a voice sample
- Development stage:
  - Modelling GMM-UBM, $\lambda_{\text{UBM}} = \{w_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$
  - MAP-adapt cohort speakers, λ_s, and extract stacked mean supervector

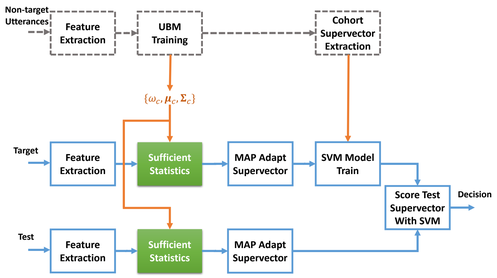

Figure 1: *Basic SVM SR System*

- Enrollment stage:
  - Mean supervectors are extracted from target speakers using MAP adaptation
  - Support vector machine (SVM) classifiers are trained using target and cohort supervectors
- Test stage:
  - Test supervectors are extracted and score against target SVM model: $y(\mathbf{X}) = \sum_{n\in\mathcal{S}} a_{C,n} l_{C,n} \mathbf{m}_{\text{test}}^T \mathbf{m}_{C,n} + b_C$, where $l_{C,n}$ denotes labels associated with support vectors

2. COMPUTATIONAL BOTTLENECK IN SUFFICIENT STATISTICS CALCULATION

- Each input feature vector, $\mathbf{x}_t$, is probabilistically aligned with each of the M UBM component densities
- Alignment and sufficient statistics calculation is performed in fast majority of SV systems
  - GMM-UBM based systems
  - Joint factor analysis systems
  - SVM supervector systems
  - i-vector systems

3. SINGLE LAYER GMM HASH SYSTEM

Hash system consists of a lower resolution hash GMM with $M_h < M$ components and parameters $\lambda_{\text{hash}} = \{w_c^{\text{h}}, \boldsymbol{\mu}_c^{\text{h}}, \boldsymbol{\Sigma}_c^{\text{h}}\}$

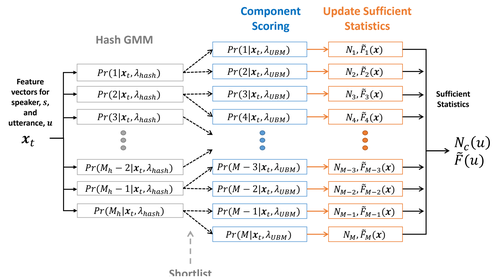

Figure 2: *Single Layer Hash System*

Shortlist mapping links components within hash GMM to components within GMM-UBM

Maximum processing reduction in alignment calculations is $\frac{\sqrt{M}}{2}$ which occurs when $M_h = \sqrt{M}$ and uniform clusters

4. HASH TRAINING VIA GAUSSIAN MIXTURE REDUCTION

Hash GMM is trained by iteratively performing Gaussian reduction

Step 1: Calculate the upper bound in between the pre-merged GMM and the post-merged GMM for every pair of components i and j within the pre-merged GMM.

$$U(i,j) \leq \frac{1}{2}\left[(w_i + w_j)\log\det\left(\boldsymbol{\Sigma}_{ij}\right) - w_i \log\det\left(\boldsymbol{\Sigma}_i\right) - w_j \log\det\left(\boldsymbol{\Sigma}_j\right)\right]$$

Step 2: Choose the pair of components with indices i and j that minimize the bound and merge the pair using to obtain the new weight, mean vector, and covariance matrix.

$$\begin{aligned} w_{ij} &= w_i + w_j \\ \boldsymbol{\mu}_{ij} &= w_{i|ij}\boldsymbol{\mu}_i + w_{j|ij}\boldsymbol{\mu}_j \\ \boldsymbol{\Sigma}_{ij} &= w_{i|ij}\boldsymbol{\Sigma}_i + w_{j|ij}\boldsymbol{\Sigma}_j + w_{i|ij}w_{j|ij}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^{\mathsf{T}} \end{aligned}$$

where $w_{i|ij} = w_i/(w_i + w_j)$ and $w_{j|ij} = w_j/(w_i + w_j)$

Hash system only needs original GMM-UBM for training

Components chosen for merging based on:

1. Components with low weights–w_i and w_j
2. Components whose mean vectors are close to each other with respect to their variances–as seen by analyzing the calculation of $\boldsymbol{\Sigma}_{ij}$
3. Components with similar covariance matrices–$\boldsymbol{\Sigma}_i \approx \boldsymbol{\Sigma}_j$

5. TREE-STRUCTURED HASH

- Each layer of tree is formed by recursively performing Gaussian reduction

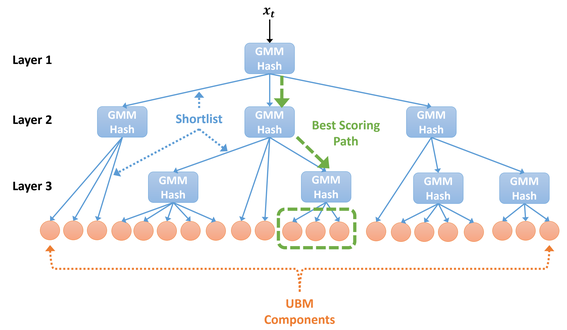

Figure 3: *Tree Structured Hash.*

- Tree structure is non-uniform due to GMM reduction merging criteria
- Each unique hash GMM in tree has M_h components
- Not all branches are traversed during testing–only those that are best aligned with input feature vector
- Maximum process reduction can be achieved using binary tree

6. EXPERIMENTS

- Evaluated with NIST 2005 SRE data–8 conversation training and 1 conversation test
- Development Corpora
  - 512 component gender-independent UBM trained with Switchboard II Phase 1, Swwitchboard Cellular Part 2, and OGI National Cellular
  - Cohorts taken from Fisher English
- Feature vectors
  - 13 MFCCs and 13 Δ-MFCCs are extracted from speech
  - RASTA processed
  - Mean/variance normalized
- Hash system was used in supervector extraction for test segments but not during system development or target speaker enrollment
- Computational reduction reported in as reduction in overall numnber of alignment calculations

7. RESULTS

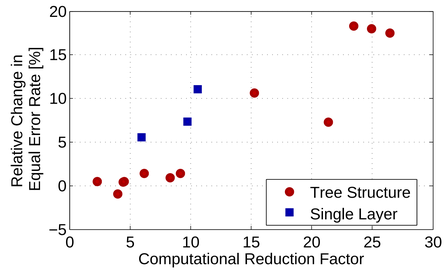

Figure 4: *Single layer vs tree-structured hash. The tree-structured hash has higher computational reduction or less degradation in accuracy for fixed computational reduction.*

Table 1: *Results of SV system using proposed tree-structured hash*

L	M_h	B	Reduction Factor	ΔEER %
9	2	1	24.93	17.97
8	2	1	23.47	18.30
4	4	1	26.46	17.51
4	4	2	4.53	0.45
3	**8**	**1**	**21.38**	**7.31**
3	8	2	8.31	0.92
3	**8**	**3**	**4.01**	**-0.94**
2	16	1	15.27	10.60
2	16	2	9.17	1.38
1	8	1	5.96	5.53
1	16	1	9.72	7.37
1	32	1	10.60	11.06

- Achieve $21\times$ reduction with 7.31% relative degradation in equal error rate (EER) with $(L, M_h, B) = (3, 8, 1)$
- Achieve $4\times$ reduction with no degradation with $(L, M_h, B) = (3, 8, 3)$

8. CONCLUSIONS

- In this paper, an efficient method for extracting sufficient statistics was evaluated in an SVM supervector SV system
- Significant processing reductions were demonstrated with small relative degradation in EER
- Demonstrated that hash-tree can be applied to previously developed SV system and previously enrolled speaker models