STBU NIST SRE 2006 System Description

Niko Brummer and Albert Strasheim

Introduction

STBU is a collaboration between 4 partners:

- Spescom DataVoice (SDV), South Africa
- TNO, Netherlands
- Brno University of Technology (BUT), Chech Republic
- University Stellenbosch (SUN), South Africa

We submit three fusions of multiple sub-systems contributed by all four partners. All submissions are run only on the primary evaluation task, both with and without unsupervised adaptation.

This document describes the details of the fusion and calibration, as well as the sub-systems contributed by SDV and SUN. Please also refer to the separate system descriptions of BUT and TNO for descriptions of their sub-systems.

We used three main kinds of systems:

- GMM, with short-time MFCC or PLP features. (SDV,TNO,BUT)
- GMM-SVM, using GMM mean supervectors as input. (SDV,TNO,BUT,SUN)
- MLLR-SVM, using MLLR speaker adaptation coefficients derived from a speech recognizer (BUT,SUN).

All systems used linear supervector-space channel compensation techniques. In the GMM case (BUT) this technique is referred to as eigenchannel MAP-adaptation. In the SVM case it is referred to as NAP (nuisance attribute projection). In all cases we used SRE 2004 and 2005 data to derive these adaptation coefficients.

SDV systems

The SDV GMM is a basic GMM system with a 512-component UBM and 24-dimensional MFCC+delta features. The features are short-time-gaussianized, but not feature-mapped. This is the same system as described in the SDV 2004 and 2005 SRE system descriptions.

We found there to be some benefit in not using our previously developed eigen-channel compensation. Rather, we essentially changed the GMM scoring mechanism to an SVM-based one. That is, we worked with 512*24 = 12288-dimensional supervectors. The channel compensation was done by projecting away 40 dimensions in supervector space. We found this to be orders of magnitude faster than our previous eigenchannel compensation and also somewhat more accurate.

We used SRE 1999-2002 for our SVM background; (extended training) SRE 2004 and 2005 for channel-space estimation; and 310 T-norm speakers from SRE 2004. We used the MATLAB function eigs() to do supervector-space PCA. We found that since eigs() employs an approximate iterative technique, that it helped to orthogonalize the eigenvectors using a singular value decomposition of the eigenvector matrix. (If the eigenvectors are not ortho-normal, they don't project away completely.)

We used the Java version of SvmLib, with a pre-computed Gram matrix. This allowed us to train each SVM speaker model in less than 1 second.

SDV contributed two systems to the fusion: A T-normed GMM-SVM-NAP run in the forward direction (train,test), as well as a reverse T-normed variant (test,train).

Fusion

All systems were fused with linear logistic regression. We had the complication that not all sub-systems were able to contribute a score for each trial, because of failure to detect speech in train or test segment, or lack of ASR transcription. This necessitated a two step fusion strategy.

- First, each system on its own was subjected to an affine calibration transformation, also trained via logistic regression. We used a logistic regression prior-weighting of 0.5 here. The training data for this calibration was all of the trials that the system could contribute of the SRE 2005 (1c4w-1c4w) trials.
- Next, scores for missing trials of each system were inserted as log-likelihood-ratio (LLR) = 0. Now all systems had valid scores for all trials and could be fused, with linear logistic regression, but this time using a prior-weighting of 0.0917 to best serve the NIST-CDET operating point.

Calibration

All three of our systems submitted scores in likelihood-ratio (LR) format.

- Two of our systems STBU-1 and STBU-2 relied purely on the affine calibration afforded by the fusion step. For these two systems, we did (somewhat arbitrarily) clip the LLR magnitude to +- 15. All that remained was to (i) threshold decisions at an LLR threshold of 2.29 and then (ii) to exponentiate LLR scores to submit LR scores.
- STBU-3 followed the fusion with a soft saturating non-linearity, called S-Cal, which is also trained with logistic regression.

For details on fusion and calibration see: http://www.dsp.sun.ac.za/~nbrummer/focal/

Submitted systems

STBU-1

We designate the unsupervised adaptation mode of this system as our primary system. This is an 11-fold fusion of:

1. GMM-SVM forward, T-normed (SDV)
2. GMM-SVM reverse, T-normed (SDV)
3. Eigenchannel GMM (BUT)
4. Eigenchannel GMM T-normed (BUT)
5. GMM-SVM T-normed (BUT)
6. MLLR-SVM (BUT)
7. GMM-SVM T-normed (SUN)
8. MLLR-SVM v1 (SUN)
9. MLLR-SVM v2 (SUN)
10. GMM-SVM T-normed, without unsupervised adaptation (TNO)
11. GMM-SVM T-normed, with unsupervised adaptation (TNO)

For the non-adaptive (n-mode) variant of this system, we simply omitted sub-system 11.

STBU-2

This is the same as STBU-1 in all respects, except that the eigenchannel GMM systems were omitted. This makes this a pure fusion of SVM systems.

STBU-3

This is the same as STBU-1, except that the above-mentioned S-Cal non-linearity was added as a further calibration aid. We found very similar, but not identical S-Cal, coefficients for the n and u modes. On our development data, APE-curve analysis showed that this brought a significant improvement in quality calibration.

Timing

Although SDV systems improved in accuracy since 2005, from about 10% EER to 7% EER, on the 2005 data, the main improvement was in execution speed. Speeds are given for a single processor 3GHz P-IV with 1.5GB of dual-channel DDR-400 RAM. (The memory access speed is very important when multiplying supervectors.)

- Each conversation side is processed up to a GMM-model in about 6 seconds.
- T-norm models are trained in somewhat less than 1s.
- Trials are recognized (using 310 T-norm models) in about 15 or 20 minutes for the whole SRE 2005 or 2006 1c4w-1cw4 test set.

SUN: Sub-systems

The University of Stellenbosch implemented SVM systems that processed supervectors obtained from the BRNO GMM and MLLR systems.

SVM systems in general

The SVM systems were implemented in Python using NumPy and libsvm 2.82. libsvm is a C/C++ and Java implementation of Support Vector Machines. Support for precomputed kernel matrices was added to libsvm in April of 2006, and we used this feature to obtain excellent performance improvements. libsvm was called from the Python code using the ctypes library.

All the SVM systems used a linear kernel. Training SVM models involves the calculation of dot products between all the supervectors presented to the SVM. These dot products make up the kernel matrix. There is a large disparity between the number of background vectors (2606 for GMM SVM, 4266 for MLLR SVM) and the number of speaker vectors (typically 1, 3, or 8 for NIST evaluations). The dot products between the background vectors can be calculated once, leaving only the dot products between the speaker vectors and the background vectors to be calculated when a speaker model has to be trained.

A further speedup can be obtained by extracting the selected support vectors from a libsvm model after training and calculating their weighted sum, in effect collapsing each model to a single supervector. The same technique can be used for the TNorm models.

Now, performing the trials for the model simply involves computing a small part of the kernel matrix ($N+m$ dot products, with N background vectors and m speaker vectors), training the SVM model, collapsing the model to one supervector (a weighted sum over a few hundred supervectors), and scoring the trials ($P+1$ dot products, with P TNorm models).

Stellenbosch GMM SVM+NAP

This systems used the GMM means from the BRNO 512-mixture, 39-dimensional feature GMM system, which resulted in 19968-dimensional supervectors. All supervectors were normalized by dividing by the standard deviation of the GMM UBM.

2606 speakers from the Fisher database were used as background speakers for the SVM. 300 from this set was also used to train leave-one-out TNorm models.

Forty NAP eigenvectors corresponding to the largest eigenvalues were estimated from 4433 segments from the NIST 2004 Extended data, spoken by 301 speakers. This yielded a 19968-by-40 adaptation matrix. All supervectors were normalized using this matrix. Some experiments were done with using more or less eigenvectors for NAP, but 40 was found to give the best results on the 2005 evaluation. The performance improvement obtained from NAP was found to increase when estimating the NAP parameters from all the 2004 data instead of using only the train and test segments from the 1side data (1774 segments).

Starting with the unnormalized GMM means of the 2006 data and the background data in binary files and the adapatation matrix already computed, this system was able to complete the 2006 1side-1side evaluation (813 models, 53672 trials) in 45 minutes. This time was spent as follows:

- 7 minutes for precomputing kernel matrix
- 5 minutes for training and checking TNorm models
- 17 minutes for training and checking target models
- 7 minutes for estimating TNorm parameters (once per test segment)
- 8 minutes for performing trials
- 1 minute for other operations

Stellenbosch MLLR SVM+NAP

Two variations of this system were implemented. In both cases, MLLR transforms from the BRNO were used:

1. CMLLR transform, 1 MLLR transform (2 transforms in total)
2. CMLLR transform, 2 MLLR transforms (3 transforms in total)

The BRNO systems generated 2 or 3 transforms, one of which was a silence transform. This transform was discarded in all cases. Each transform was made up of a block-diagonal matrix containing three 13-by-13 matrices and a 39-dimensional bias vectors, yielding 546 components per transform, or 1092-- and 1638-dimensional supervectors.

Rank normalization was applied to the supervectors, with normalization parameters estimated from 4266 segments from the NIST 2004 Extended data, spoken by about 310 speakers. These same segments were used for the SVM background. 300 from this set was also used to train leave-one-out TNorm models, but TNorm was found to reduce performance.

Fifteen NAP eigenvectors corresponding to the largest eigenvalues were estimated from 4159 segments from the NIST 2004 Extended data, spoken by 301 speakers. This yielded a 1092-by-15 or 1638-by-15 adaptation matrix. All supervectors were normalized using this matrix. Some experiments were done with using more or less eigenvectors for NAP, but 15 was found to give the best results on the 2005 evaluation.

We started with the unnormalized supervectors of the 2006 data and the background data in binary files and the adapatation matrix already computed. The 2-transform system completed the 2006 1side-1side evaluation (812 models, 53522 trials) in 16 minutes. The 3-transform system completed the evaluation (812 models, 53631 trials) in 18 minutes.

There is a reduced number of trials and models compared to the GMM SVM system because BRNO's MLLR system did not produce supervectors for all the required train and test segments.