Beyond the Equal Error Rate -
About the inter-relationship between algorithm and application

Renana Peres

Comverse Technology
renana_peres@persay.com

Abstract

Speaker verification technologies have many commercial applications, such as direct banking, cellular transactions, credit card operations and E-Commerce. Voice based verification can answer the need for a secure, friendly and cost effective authentication tool required by the finance, commerce and Telecommunication markets.

Introducing an operational large-scale system to the market requires much more than a good algorithm. Several design issues should be considered, such as:

How to retrieve the audio from the telephony network? What is the optimal way to store and maintain the voice signatures? How to receive claimed identity? Is log likelihood a meaningful score?

The development process opens a wide range of subjects for algorithmic research. Among them are: time evolution of speaker models, decision mechanisms, effective scoring, and new ways for constructing world models.

Algorithmic research and system development cannot be done independently. Continuous joint work is necessary in order to have successful operational systems, which will make speaker verification the natural authentication means in remote services and transactions.

The paper reviews the inter-relationship between algorithmic research and system development based on the experience from the speaker verification product of Persay Ltd. We describe the main problems during the system design process, and discuss the alternatives for solution. A list of research problems, derived from the implementation process is presented.

1. Introduction

Speaker verification technologies have a large variety of commercial applications, mainly in the field of authentication solutions for ***remote services***. With the development of telephony and Internet infrastructures, an increasing number of services are provided to users by remote access. Many of us have an account in a direct bank, shop through the telephone and the Internet, use calling cards, and roam between countries with our GSM. The remote services market is estimated today by over 1 billion US$ a year, and increases on average by 20% each year.

Remote services are by nature vulnerable to fraud. The service provider has to verify that the true user is the one who receives the service. Passwords, PIN codes and verification questions (e.g. your mother's maiden name) are not sufficient. The industry is looking for authentication tools, which will be secure, easy to operate through telephony and Internet networks, user-friendly and cost effective.

Voice based verification, using ***vocal password*** or ***free speech*** can be the solution in many applications of remote services.

In order to produce a reliable large-scale voice verification system, one needs to cope with various issues, which go far beyond the bare algorithmic performance. The design and development process opens a new range of research problems, which are necessary for successful implementation, and are seldom treated by researchers.

In this paper we investigate some of the issues raised when converting a set of speaker verification algorithms into an operational system. The work is based on the development process of Orpheus, the speaker verification system of Persay Ltd., from the Comverse group.

Chapter 2 describes optional operational scenarios where a speaker verification system is used in a real-life environment. Chapter 3 illustrates the general architecture and main modules of a speaker verification system. In Chapter 4, the design considerations for each module are discussed, and alternatives for solutions are suggested. The focus is not on the solution chosen, but on mapping the possible alternatives according to application types. In each subject, new research problems, which have emerged from the process, are presented.

2. Operational Scenarios

Speaker verification technologies can be used in many applications, in a variety of operational scenarios. We present here two representative operational scenarios: One for ***free speech*** (using Text independent technology) and one for ***vocal password*** (text dependent technology). These scenarios can be applied in Call Center, Telecommunication and Internet environments.

In free speech mode, a conversation is carried out between the user and another person, for example, the agent in a Call Center. The speaker verification system works behind the scenes, samples the caller audio, compares it with a voice signature which was previously collected and trained, and indicates whether or not the speaker voice matches the claimed identity. The process is automatic, transparent to the caller, independent of content, and in many cases, also of language. The enrolment process is built-in: the samples for the voice signature are collected automatically during the first few calls of the user .

When the caller interacts with a machine (such as an IVR system), the vocal password mode is used. The caller chooses a vocal password, the system compares it with the voice signature, and allows or denies access. In cases of denial, the caller may have another trial. The voice signature is collected from a special enrolment session, where the password is repeated several times.

The figure below illustrates the two scenarios.

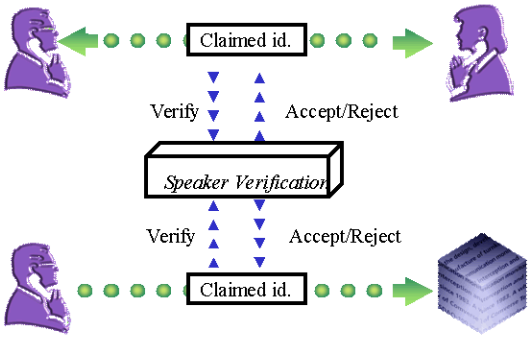

Figure 1: Optional operational scenarios for voice-based verification.

3. General Architecture of a Speaker Verification System

Every speaker verification system has the following components, as described in Figure 2:

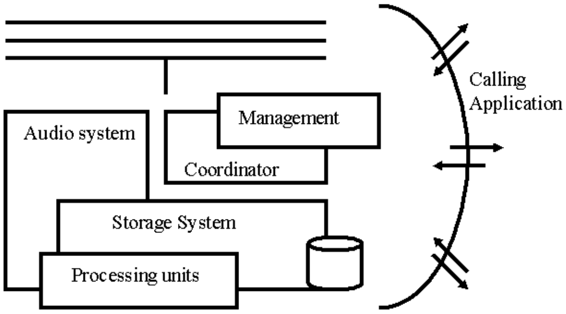

Figure 2: Typical architecture of a speaker verification system.

1. **Processing units** - The algorithmic core of the system, responsible for performing the algorithmic operations of the recognition process: ***calibration***, ***training***, and ***verification***.
2. **Audio system** - Responsible to retrieve the audio from the telephony of Internet network, transfer it to the system, and perform initial processing, if needed.
3. **Storage system** - The database which stores and maintains the ***voice signatures*** (enrolment audio and speaker models). It also stores the mapping between the claimed identity and the voice signatures.
4. **Coordinator** - Initiates and manages the missions, receives the results from the processing units, and transfers them to the calling application.
5. **Management** - Monitors the system status, provides tools for maintenance, and reports about system performance.
6. **Calling application** - The entity in the service provider environment, which interfaces with the speaker verification system.

The relationship between modules and the exact mission flow are derived from the operational scenario and are influenced by operational constraints and end user needs.

4. Implementation Issues

Integrating the algorithmic core into a large-scale operational environment requires answers to many design issues. The design process effects the algorithmic research and opens new research challenges, which rarely arise when developing algorithms for lab purposes.

4.1. Audio

The main task of the audio system is to retrieve caller voice from the telephony or Internet network, and transfer it to the processing units. In vocal password applications, the task is simple - samples are delivered as audio files directly from the IVR system. Free speech mode is a different story. Verification is performed on-line, in parallel to the call. Therefore, the audio system has to connect physically to the audio trunks, continuously pull the audio from all the concurrent calls, and transfer each sample to the relevant recognition mission. Volumes of audio are large: as example, the typical Call Center consists of 100 agents, i.e. 100 concurrent calls. A typical Telecommunication application usually involves 10 digital trunks, i.e. 300 concurrent calls. Handling such volumes in real time, while keeping constant synchronization with the claimed identity indications arriving from the application module, is an extremely complex task.

Audio retrieval raises several topics for research. First, the two sides of the call are not always separated. Digital trunks are always 4 wire, however, in many applications audio accumulation is done from 2 wire channels, where two sides are already summed. In those cases, speaker segmentation and separation algorithms should be applied. Simplistic separation methods cannot be used since in many cases, there are more than two participants in the conversation (the call is transferred between agents etc.).

Another problem is related to the VOX. The incoming audio contains long periods of silence, non-speech segments, DTMF tones, and many other noises. Cleaning and filtering the audio in an early stage can improve algorithmic performance and save significantly in transfer volumes, saving which is immediately translated into reduction in system size and costs . One has to find the appropriate kind of VOX, which retains only the required info, but maintains the long-term characteristics of the signal. In vocal password applications, speakers often insert speech sounds, which are not related to the password, such as "eeh", "aah" etc. Finding a method, which filters those segments, and retains only the password samples, is an exciting research problem, and can improve performance significantly.

4.2. Storage

The storage system stores all the objects, which should be saved between missions. It includes the voice signatures, world models, accumulated performance data, etc.

Since the storage contains sensitive, private information about users, many organizations require that the data will be maintained by them, and will not be part of the speaker verification system. Sometimes, the verification systems are distributed between sites, and are all served by a single central database. System architecture should be flexible to support any required configuration, and storage modules should be encapsulated, so database configuration will be completely transparent to the other modules.

An important question is what exactly should be stored? The main stored objects are the voice signatures. A voice signature usually contains the enrolment audio (2-3 sessions in free speech applications, 1 session in vocal password), and the speaker model. Storing the audio of the voice signatures is required for refinement and re-training, and for debugging purposes. In large systems, storage volumes become a critical variable: One minute of raw audio in PCM format has the size of 500Kbytes; 2-3 sessions are required for the signature. A system serving 500,000 users should store hundreds of gigabytes just for voice signatures. All this volume should be back-upped regularly, copied and re-stored for redundancy, and transferred quickly to the processing units when needed. Standard maintenance mechanisms, which usually come with commercial database systems, cannot be used since most commercial databases are not able to deal efficiently with audio.

To provide robust and reliable performance, voice signatures and speaker models should be continuously maintained and updated. Faulty voice signatures should be identified and repaired in an automatic process. New sessions should be added from time to time to the signature, to keep track of changes in the speaker voice, and the addition of new sources (handsets, cellular handsets etc.). When an improved algorithmic version is installed, all signatures should be re-trained.

Voice signature maintenance opens new problems for algorithmic research. We list here several of them: Being able to add new sessions to the voice signature and re-train without keeping the previous audio can be crucial for operating large scale systems. How to identify a faulty session in a signature? How often should voice signatures be updated? Is there a difference in this sense between text dependent and text independent voice signatures? All those are problems that require some level of solution before an operational installation could be performed.

As mentioned above, the heavy storage load is caused by the audio of the voice signatures. Speaker models are smaller and range between 5-20Kbyte per signature. In applications like smart cards, even this is too much, and signature size should not exceed the limit of one kilobyte. New and interesting modeling methods can emerge from this constraint.

4.3. Recognition Phases

The recognition process can be divided into three main phases: Calibration, enrolment, and verification. ***Calibration*** is the initialization of the system: World models are accumulated and constructed, general system constants are determined. ***Enrolment*** is the stage of accumulating the sessions of the voice signature, and generating the speaker models. ***Verification*** is the stage of presenting a claimed identity, and checking it against the voice signature.

Usually the calibration phase is performed once as part of system installation. Enrolment and verification are performed continuously, as new subscribers join and use the service.

4.3.1. Calibration

The exact nature of the calibration stage depends on the type of the algorithm used in the system. Let us take as an example a free speech application using world models and tuning curves (inter speaker vs. intra speaker likelihood ratio distribution) for decision. In lab experiments, where pre-collected databases are used, part of the database is saved for world models and tuning. In operational systems, data for calibration is collected from real conversations going through the system. Since world models usually require a lot of audio, all system modules should take care of accumulating, storing and processing hundreds of minutes of audio. This is a heavy load on the system, and a source for errors (faulty audio, processing errors etc.). Any error occurring during this process will influence the enrolment and verification missions during the whole life-cycle of the system. Therefore, system designers should provide elaborate error tracking and repairing mechanisms.

In many standard databases, sessions are labeled according to source (media), gender, etc. This is of course not the situation in most of the real-life scenarios. If performance depends on pre-defined similarities between the voice signature and the world model, methods for unsupervised clustering of calibration data should be found.

As implied by their name, world models are a representative ensemble of speakers in a given environment. However, this environment is not static and is constantly changing - new handsets, new compression methods, change in the wireline / wireless call ratio, all these change the characteristics of the audio environment. Exploring the effects of the change, and finding methods for on-line evolution of world models, with minimum interference on the system ongoing operation, is an interesting research subject.

An almost uninvestigated subject deals with the role of calibration in text dependent applications. In the most general type of a vocal password application the users can choose any password they like, and algorithms cannot assume anything on the type of password. How to build world models and tuning data for a text dependent application, where data of impostors saying the user's password is not available? How to do this without any language-based information? Many unsolved problems are waiting to be answered by an intensive algorithmic research.

4.3.2. Enrolment

The enrolment phase is composed of two stages: data collection and training. ***Data collection*** is the stage where sessions of the speaker are collected for the voice signature. ***Training*** is the algorithmic processing of generating the speaker models.

A necessary condition in every large-scale application is that the data collection stage will require minimal, or no user involvement. In free speech applications, voice signature is created from the first few calls of the user. The user performs the transaction, unaware of the fact that the audio of his call is transferred to the speaker verification system for building his voice signature. In vocal password applications, usually one special data collection session is allowed. During this session, the user repeats his password for several times.

During the data collection sessions, the service provider uses alternative methods to verify the caller identity, in order to secure the transaction and make sure that the signature will not contain impostor audio. This extra-verification has costs for the service provider, therefore the data collection stage should be reduced to the minimum.

Generating representative, robust speaker models under such constraints is probably the most difficult task for the algorithmic research. The audio collected during the data collection sessions simply does not represent the variety in the speaker's voice and in the audio environment. Very few studies were done so far on mixed-source signatures. What happens where voice signature comes from wireline telephone and test data come from wireless? Is a voice signature composed of 2 or 3 sessions robust enough to represent a speaker, who has two types of cellular phones, uses his hands-free mobile from the car, but sometimes calls from home? The research community does not even have good enough databases to start and explore those questions.

One of the possible solutions to the enrolment problem is gradual evolution of the voice signatures with time. After the initial training of the voice signature, audio from subsequent calls of the user can be added to the voice signature, and a new speaker model is created. What are the criteria for adding a new session to a voice signature - if it is done on the basis of recognition performance, what should be the thresholds? Choosing a low threshold can cause the insertion of an impostor session, while choosing a high threshold might not lead to the variety of sources we are looking for.

Valuable information for controlling the enrolment process could be achieved if there was a way to create a "score", which measures the quality of the voice signature. Such a score could use as the criterion for improving the voice signature, or be combined with the verification decision (verification based on a low quality signature will have lower confidence level). An efficient signature score can help the service provider in handling problematic cases. For example, if the score would be sensible enough to detect "goats" (i.e. speakers with a consistently high false rejection rate) from their voice signatures, service providers can route those users to other authentication procedures, avoiding the inconvenience of false rejection of true subscribers.

Failures due to user mistakes, faulty audio or system load can happen during enrolment. In free speech systems, where the user is not involved in the enrolment, faulty audio is simply replaced with another session, and the signature can be re-trained. In vocal password applications, failure of the training might require another data collection session, which causes inconvenience for the user and for the service provider. Therefore, a system should be designed to perform the training during the data collection session, while the user is still on the line. If there is any failure, the user is asked for additional repetitions. This requirement for on-line training, instead of training as a background low priority process, influences system design and the interfaces between the audio system, the processing units and the coordinator module.

4.3.3. verification

A speaker verification system is busy most of the time in verification missions. The user's claimed identity is verified by comparing the incoming call with his voice signature. Based on verification results, the access to services is approved or denied.

Any service system, which involves voice-based verification, must provide each subscriber with a unique identifier, which will use as the claimed identity for verification. Such unique identifiers can be bank account no., telephone no., name etc. They can be provided to the system from several sources: CLI numbers are used when identifier is the caller telephone no. In many cases, an IVR system, either DTMF or speech based is involved.

In vocal password applications, the user is asked to say his password, and in case of ambiguity or rejection, he may have another trial. Usually, up to 3 trials are given. The system should be designed to support multiple trials: the coordinator should keep the verification mission open and direct to it the audio of all the trials of the user. The algorithms should be able to indicate whether another trial is required. The verification algorithms can share information between trials: instead of treating them as independent trials, each trial can use the audio and results of the previous trials to improve performance.

In free speech systems, verification is done in parallel to the conversation, and results can be provided in any time point during the call. The system should support result-update policies such as update in fixed time intervals, update when a certain score is reached, and update upon request.

Major design issues during the verification phase deal with decision and scoring. Applications usually require an accept/reject decision, accompanied by a score, reflecting the confidence level of the decision.

The standard decision method is setting thresholds on the likelihood ratio, or on the distribution of likelihood ratios, as described in Figure 3. Thresholds are dynamic and depend on the risk in the transaction: transferring large amount of money between accounts will have a different cost than a balance inquiry. The service provider must have full flexibility in determining decision thresholds: in large-scale systems, each percent of false rejection means lot of unsatisfied users, and each percent of false acceptance can cause heavy losses. The system should be able to perform the dynamic translation between the working point, set by the calling application in terms of false acceptance / false rejection, and the likelihood ratio, provided by the algorithm.

Another popular score is the posterior probability. The posterior probability is the proportion of correct answers among all the experiments with the same likelihood ratio. One could say that it reflects the confidence of a single trial. Decision can be made by setting upper and lower thresholds on the posterior probability. This will provide three-state decision: accept, reject, and undecided.

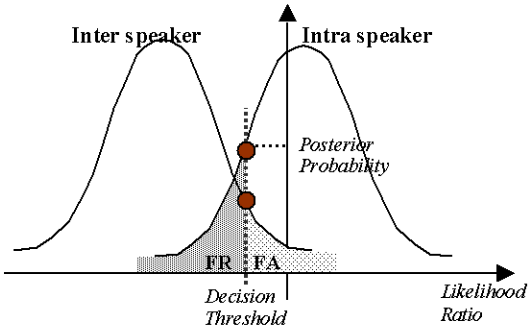

Figure 3: Threshold on the likelihood ratio is translated to false acceptance and false rejection

A decision procedure cannot be completed without considering prior probabilities. Without any other information, the assumption is of prior 0.5 for a real speaker and an impostor. In real-life applications, the prior probability for impostor is very low, however the cost of accepting an impostor is high. The right factor to use is the risk, i.e. the product of the prior, with the error cost function. Any detail the application has about these two factors can help to improve the accuracy of decisions and scores.

4.4. Management

All the large-scale systems require management and technical maintenance tools. A speaker verification system requires specific management tools, in addition to the standard ones. The service provider should have all the information as to the current status of the system, no. of verification and training missions, loads, and system availability. The system should provide information about the status of voice signatures: how many voice signatures are in the data collection stage, how many sessions were collected for each signature etc.

The service provider should be aware of any rejection case by the system. Both possibilities, false rejection of a true subscriber or a rejection of an impostor, require special treatment from the service organization.

The system should provide tools to measure its performance. For this, one needs to know the real identity of the caller. This is not always possible, but in many cases, voice based verification is used in addition to other authentication methods, and results can be compared between systems. On-line feedback from the calling application as to the correctness of the system results is a valuable information and can be used by the various system modules to detect errors and to improve performance.

5. Conclusions

In this paper, we described some of the design issues and algorithmic problems which one has to face when introducing an operational system to the market.

System designers have to answer questions concerning system architecture, interfaces, telephony, transfer volumes, user behavior, and service issues, which go far beyond the algorithmic core.

The algorithmic core is influenced and modified by the requirements and constraints of the application. The algorithms should provide answers to questions that are never dealt with in a lab environment. We described some of them throughout the paper.

Algorithmic research and system development cannot be done independently. System developers should find ways to optimize the advantages of the algorithms and to compensate for their limitations. Researchers should learn more about the nature of applications and the research problems they raise. Without the challenges set by the real application world and real-life data, algorithmic research in speaker recognition will never go beyond finding a new set of features, of a another classifier which improves the EER in few percents.

Continuous joint work will lead to a more exciting research, and better operational systems, which will make speaker verification the natural authentication means in remote services and transactions.