ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition

Jean-François Bonastre1, Nicolas Scheffer1, Driss Matrouf1, Corinne Fredouille1, Anthony Larcher1, Alexandre Preti1, Gilles Pouchoulin1, Nicholas Evans1,2, Benoit Fauve1 and John Mason2

1Laboratoire d’Informatique d’Avignon (LIA), UAPV, France 2Speech and Image group, Swansea University, Wales, UK

1firstname.name@univ-avignon.fr, 2{b.g.b.fauve.191992,J.S.D.Mason}@swansea.ac.uk

Abstract
This paper presents the ALIZE/SpkDet open source software for text independent speaker recognition. This software is based on the well-known UBM/GMM approach. It includes also the latest speaker recognition developments such as latent factor analysis and unsupervised adaptation. Discriminant classifiers such as SVM supervectors are also provided, linked with the nuisance attribute projection. The software performance is demonstrated within the framework of the NIST’06 SRE evaluation campaign. Several other applications like speaker diarization, embedded speaker recognition, password dependent speaker recognition and pathological voice assessment are also presented.

1. Introduction
Indicated by the growing number of participants in the international NIST speaker recognition evaluations (SRE) [1], text-independent automatic speaker verification (ASV) has experienced an increasing interest in recent years. At the same time meaningful improvements in performance have been achieved with error rates roughly halving over the last three NIST SRE campaigns. These two phenomena are not directly linked since the best performances invariably come from a limited number of sites, mostly with previous experience in such campaigns. And any exceptions to this observation tend to relate to site combinations where perhaps one or more of the partners have previous SRE experience. The key point here is the high level of research and technology investment required to reach and remain close to the ever moving state-of-the-art. In this context, combined effort across sites can clearly help; the simplest and most common example of this is the sharing of system scores with score-level fusion. At the other extreme of such cooperation is open source software, the potential of which is far more profound.

This paper presents an open source software: the ALIZE/SpkDet packages. The realization has taken place in the context of the open source ALIZE toolkit [2, 3] developed by the LIA in the framework of the French Research Ministry Technolangue programme. ALIZE comes from the collaboration done around the ELISA consortium [4], which grouped the efforts of several European laboratories in order to participate in NIST evaluation campaigns (mainly NIST-SRE and NIST Rich Transcription evaluation) from 1998 to 2004. ALIZE/SpkDet is a package within ALIZE tailored specifically to ASV; ALIZE/SpkDet was one of the chosen reference systems in the framework of the BioSecure Network of Excellence; it was also used by several institutions for NIST’06 SRE. Feature extraction comes from SPro [5]. All these systems are distributed through open source licences.

Recent key developments include support vector machines (SVMs) [6], the associated nuisance attribute projection compensation (NAP) [7], and factor analysis (FA) [8, 9]. These developments stem mainly from a collaboration between the LIA and the University of Wales.

This paper presents an overview of the ALIZE/SpkDet software. After a short technical description in Section 2, Section 3 presents a demonstration of the toolkit performance, using the latest NIST SRE database. In Section 4, some work concerning unsupervised adaptation of the client models are described and evaluated using the NIST’05 and ’06 SRE framework. Even if the software is mainly designed for text-independent speaker recognition, several other applications use ALIZE/SpkDet. Section 5 presents some examples of these developments. Finally, Section 6 concludes with ideas on the immediate future of the project.

2. Overview of ALIZE/SpkDet
The ALIZE project was initiated by the ELISA consortium under the responsibility of the LIA and initially funded by the French Research Ministry Technolangue program during 2003/2004. The project was continued using LIA and BioSecure funds. ALIZE was also retained for two French National Research Agency (ANR) projects, BIOBIMO and MISTRAL, allowing support until the end of 2008.

2.1. ALIZE main objectives
The main objectives of ALIZE are:

• to propose a toolkit facilitating the development of new ideas, with a (proven) state-of-the-art level of performance;
• to encourage laboratories to evaluate new proposals using the toolkit on both standard databases and protocols such as the NIST SRE evaluations;
• to help the understanding of speaker recognition algorithms (like EM training, MAP adaptation or Viterbi algorithm), parameters and limits;
• to test new commercial applications;
• to facilitate the exchanges and knowledge transfer between the academic laboratories and between academic laboratories and companies.
Thought the MISTRAL and BIOBIMO projects, some new functionalities are currently developed, like client/server or embedded architectures, text dependent speaker recognition systems and multimodal user authentication systems.

2.2. An illustration of ALIZE specificities

ALIZE is open source software developed in C++ following an object oriented UML method. The general architecture of the toolkit is based on a split of the functionalities between several software servers. The main servers are the feature server, which manages the acoustic mixture, the mixture server which deals with the models (storage, modification, tying of components, saving/reading...), and the statistics server which implements all the statistical computations (EM-based statistic estimations, likelihood computation, viterbi alignment, etc.) This architecture presents several advantages:

- Each server is accessible thanks to a very short list of high level functions and the low level functions such as memory management are usually hidden from the user;
- Each server is optimized and updated separately (thanks to this relative independence, developing new functionalities inside a new server is also very easy);
- Several instances of the same server could be launched at the same time (particularly useful for multi-stream or multimedia systems);
- The user-code presents the same structure, organized between the main servers, helping the source code development and understanding;
- Finally, the software architecture allows easily to distribute the servers on different threads or computers.

2.2.1. Data reading and synchronization

All the acoustic data management is delegated to the feature server. It relies on a two step procedure: feature server initialization and a reading call for accessing to each feature vector. The reading of an acoustic stream is performed frame by frame following a feature loop as illustrated in figure 1. The user synchronizes the reading process thanks to the reading calls. Therefore, the same source code is employed for off-line (file based) processing and for on-line (open microphone) processing (the only difference occurs in the server configuration). Accessing at different time instants in the audio stream (go backward or forward, go to frame \( x \)) is also allowed. Memory management is achieved according to the two simple rules:

- the user defines the size of the data buffer, in time, from unlimited to one frame. If the buffer shows a limited size, while the user requests a frame out of the buffer, the server will return a ‘frame non available message’ (the same message is sent if no frame is available during open microphone mode);
- the user defines the memory footprint for the data server. An integrated buffering service is provided for accessing very large datasets (several hundred hours of speech).

2.2.2. EM/ML world model training

For training a GMM world model, the user has to initialize three servers, the feature server for the data, the mixture server for managing the Gaussian components (and mixture models) and the statistics server for estimating the statistics. Figure 2 shows the skeleton of an example EM training procedure, beginning from scratch (the model is randomly initialized by default but the initialization could be easily modified).

2.2.3. EM/MAP speaker model estimation

For deriving a speaker model from the world model using a MAP adaptation algorithm, the user builds exactly the same program as the previous one. Only two differences have to be highlighted: the client model is initialized as a copy of the world model and the final model (at each iteration) is the result of MAP(), a function involving both the world model (the a priori knowledge) and the statistics estimated on the client training data. Implementing some variants of the MAP algorithm will only take place in this MAP() function. This process is illustrated Figure 3.

2.2.4. Score computation

The score computation follows the same structure as the two previous programs. Figure 4 shows an example of the (log) likelihood computation for several client models, using a n-top Gaussian computing. For top Gaussian computing, the user needs only to set an optional flag (DETERMINE_TOP_DISTRIBUTBS) during the corresponding call for memorizing the winning components and to set the flag to a different value (USES_TOP_DISTRIBUTBS) for using it for the other calls. An implicit component tying is also implemented and helps to save both computational time and memory.
Discriminant classifiers like the SVM were proposed during the past years in several works as in [10, 11, 12, 13]. These classifiers are usually applied to GMM supervectors. A GMM supervector is composed of the means of a classical GMM system, as initially proposed by [13]. Libsvm library\(^3\) is used for the basic SVM functionalities.

Session variability modeling

Some of the most important developments in ASV over recent years relate to strategies that address the “mismatch factor”. ALIZE/SpkDet includes a set of functionalities following the Factor Analysis proposed by [8] and the Nuisance Attribute Projection proposed by [7]. In these approaches the goal is to directly model the mismatch rather than to compensate for their effects as it was done with H-norm and T-norm. This involves estimating the variabilities from a large database in which each speaker is recorded in multiple sessions. The underlying hypothesis is that a low dimensional “session variability” subspace exists with only limited overlap on speaker specific information. Both Factor Analysis and NAP were developed inside ALIZE/SpkDet, using the SVDLIBC\(^4\) toolkit for the singular value decomposition.

3. Performance of the software

3.1. Experimental protocol

The male part of the NIST’05 primary task (1.conv4w-l.conv4w) is used for development (DevSet). For this condition, one side of a 5-minute long conversation is available for testing and the same amount for training. All background training data, for the universal background model (UBM), T-norm [14], NAP, and FA come from the NIST’04 database. This procedure leaves the NIST’06 free for validation. The final comparisons are made on the NIST’06 core (required) condition which includes multiple languages (rather than the English only common condition).

Performances are assessed using DET plots and measured in terms of equal error rate (EER) and the minimum of the decision cost (minDCF). The cost function is calculated according to NIST criteria [15].

3.2. Description of the systems

This paragraph presents several systems in order to illustrate the large scale of techniques embedded in the software as well as the level of performance achieved using ALIZE/SpkDet (more details on the different systems are in [16]):

- GMM baseline (GMM). A classical GMM system is developed using ALIZE/SpeakerDet. The features are based on 19 linear filter-bank derived cepstra (computed using the open source SPro toolkit[5]). The feature vector is composed of 50 coefficients, 19 static coefficients, 19 delta and 11 delta-delta and the delta energy. A classical energy-based frame pruning system is applied before to normalize the recordings, file-by-file (cepstral mean subtraction and variance normalization). Feature mapping is also applied. The UBM model size is 512 (with diagonal covariance matrices) with the speaker models adapted from this UBM via a MAP procedure. This GMM baseline system is also used for the other systems.

- Gmm Supervector Linear kernel (GSL). The GSL system uses a SVM classifier applied on GMM supervectors. The supervectors are taken directly from the GMM baseline system, giving in our case a vector size of 512*50.

- Gmm Supervector Linear kernel + Nuisance Attribute Projection (GSL-NAP). GSL-NAP corresponds to a GSL system using the nuisance attribute projection technique [7] in order to deal with intersession variabilities

- Symmetrical latent Factor Analysis (SFA). Recently new approaches have been proposed by Kenny [8] with factor analysis (FA) in a generative framework. This approach was implemented into our software using an original symmetrical approach presented in [17]
Figure 5: Det plots for GSL-NAP, SFA and their unweighted linear fusion, rank=40, T-norm (devSet)

### 3.3. Performance on NIST SRE 2006 database

<table>
<thead>
<tr>
<th></th>
<th>male 05</th>
<th>male 06</th>
<th>all 06</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCF</td>
<td>EER</td>
<td>DCF</td>
</tr>
<tr>
<td>GMM baseline</td>
<td>4.37</td>
<td>8.07</td>
<td>3.94</td>
</tr>
<tr>
<td>GSL</td>
<td>2.79</td>
<td>8.02</td>
<td>3.37</td>
</tr>
<tr>
<td>GSL-NAP</td>
<td>1.62</td>
<td>5.28</td>
<td>2.07</td>
</tr>
<tr>
<td>SFA</td>
<td>1.94</td>
<td>4.38</td>
<td>2.17</td>
</tr>
</tbody>
</table>

Table 1: Performance in EER(%) and minDCF(x100) for the GMM, GSL, GSL-NAP and latent factor analysis GMM (SFA) on 05 development set, male’06 and male and female combined ’06 validation sets

Performance values are presented in Table 1 for all the independent systems, for the development set and for the different validation sets (issued from NIST-SRE 2006). The results confirm that NAP and SFA techniques bring a significant improvement compared to the classical GMM system. Compared to the official results of NIST SRE 2006, available in [18], it is clear that ALIZE/SpkDet are among the few systems able to achieve an EER in the region of 5%.

Figure 5 shows the results of an unweighted linear fusion between the 2 best systems namely SFA and GSL-NAP\(^3\). The improvement shows their complementarity, even if the two systems share the same acoustical representation, the same GMM parameters (UBM) and the same session variability parameters.

### 4. Unsupervised adaptation

A classical solution in order to improve the performance of a speaker recognition system is to increase the amount of information used to train the client model. However, this solution depends on the availability of such training data. Unsupervised adaptation of the client models is a good solution to this data availability problem. In unsupervised adaptation, the client model is updated online, using data gathered from the test trials. The different approaches to unsupervised adaptation proposed in the literature rely mainly on a decision step to decide if a test trial belongs to the targeted speaker [19, 20, 21, 22, 23]. If the test trial is considered as client, it is used to either retrain the corresponding client model or to adapt it. The main drawback of such techniques remains the difficulty to set a decision threshold for selecting the trials: the performance gain relies on the number of client test trials detected, a false acceptance (an impostor trial is accepted as a client one) degrades the speaker model.

The work presented in this Section was done in collaboration with Thales Communication and is presented in detail in [24]. It aims to avoid the decision threshold problem by negativising the need for this early decision step. In this approach, the speaker model adaptation is applied to each test trial, even if it does not belong to the client but to an impostor. This “continuous adaptation” method relies on a confidence measure on the matching between the test trial and the target speaker, used to weight each test information during the speaker model adaptation process.

#### 4.1. Confidence measure estimation

The confidence measure is the estimation of the \emph{a posteriori} probability of a test trial belonging to a given target speaker. This \emph{a posteriori} probability is computed using two score models, one for the client scores and one for the impostor scores. Each score distribution is modelled by a 12 component GMM learned on a development set. The confidence measure is then computed using the WMAP approach [22, 25]. To avoid the problem of the re-estimation of the WMAP function after each adaptation step, we use only the initial target model, learned on a single session recording, to compute the score of the test trials.

#### 4.2. Proposed adaptation function

The proposed adaptation function relies on the classical MAP algorithm [26], where only the mean parameters are updated. The empirical statistics are gathered from all the available data using the EM algorithm (initialized with the background model and maximizing the ML criterion). The statistics are then combined using the following rules:

- The statistics gathered from the initial voice excerpt used to train the target speaker model is associated to a confidence measure equal to 1;
- The statistics gathered from the different test trials are associated with the corresponding confidence measure;
- The empirical means and the corresponding occupancies are computed for each Gaussian component of the GMM, using all the EM statistics weighted by the corresponding confidence measures.

Finally, the adapted means (\(\mu_{\text{adapt}}\)) for each Gaussian component (\(i\)) are computed using the following formula:

\[
\mu_{\text{adapt}}(i) = \frac{\mu_{\text{emp}}(i) \cdot \mu_{\text{map}}(i) + \mu_{\text{ubm}}(i) \cdot (1 - \mu_{\text{map}}(i))}{\mu_{\text{map}}(i) + (1 - \mu_{\text{map}}(i))}
\]

\(\mu_{\text{emp}}(i)\) and \(\mu_{\text{ubm}}(i)\) are the empirical and the background means for each Gaussian component respectively.

#### 4.3. Experiments and results

All the experiments presented here are performed based upon the NIST’05 and ’06 databases, all trials (det 1), 1conv-4w 1conv-4w, restricted to male speakers only. The baseline GMM system is the one described in Section 3. In addition, a version of the latent factor analysis is used at the feature level: in this case, the channel compensation is applied to each feature set instead of the feature mapping. Figure 6 presents the results for the adapted system and the baseline on the NIST’05 database, for feature mapping and the LFA channel compensation techniques.
The results demonstrate the potential of the proposed method as it reaches a significant 27% DCF relative gain (and 37% in terms of EER) with the feature mapping (FM). When the LFA-normalized features are used, the DCF gain is about 20% (and 12.5% for the EER). The gain is smaller in this case, as expected because LFA is known to perform better channel compensation than FM, reducing the influence of the unsupervised adaptation on the channel effects.

In Figure 7, we attempt to analyze more precisely the behavior of our method. This figure shows the performance in terms of min DCF of the adapted system for each newly added test trial ($n$ denotes the number of test trials added to the system).

Results for the adapted system and the baseline are provided in Figure 8, for the NIST’06 database and using both FM and LFA. The results are disappointing compared to those of NIST’05. On this database, the unsupervised adaptation method introduces a significant loss. Different factors could explain this unexpected result with the most relevant thought to be:

- The percentage of impostor trials versus target trials differs between the databases. Whilst 9.0% of target tests are proposed in NIST’05, only 7.1% are present in NIST’06. Even if the difference seems quite small, it corresponds to 21% less target data for the 2006 database. Moreover, the number of test trials by target speaker is also smaller for NIST’06.
- When looking at the target and impostor score distributions of the baseline system we observe that a large part of the errors comes from a small percentage of the impostor trials which obtained a very high score. This phenomena is significantly higher for the NIST’06 than for the NIST’05 database. It shows the obvious dependence between the baseline system and the adaptation process behavior.

5. Other applications

5.1. Embedded password dependent speaker verification

For embedded speaker recognition, the main issues are usually the memory and computational constraints. But realistic embedded applications generally require short utterances, few training data and very noisy environments, giving degraded speech quality. In order to deal with these application constraints, we moved from a text-independent speaker recognition scenario to a text-dependent authentication using passwords that were chosen by the clients. This section presents the LIA_{EBD} package based on ALIZE and dedicated to such embedded applications.

5.1.1. Description

Our approach merges a state of the art GMM/UBM system and a semi-continuous HMM architecture, following the ideas developed in [27] with a DTW algorithm. Hierarchical three-step models are created where a lower level model takes some of the parameters from the immediate upper level model. The top model is a classical UBM, the second one is a quite classical text-independent speaker model. It is adapted from the
UBM, updating only the mean parameters using a MAP procedure. In LIA_EBD, only few UBM components are adapted and saved, in order to save both memory and computation time (other mean parameters, weight parameters and variance parameters are taken from the UBM model). In the lower level, a semi-continuous HMM (SCHMM) is built in order to take into account the phonetic information and the duration information. Each state of the SCHMM is linked to a GMM distribution. The SCHMM state-dependent distributions are adapted from the corresponding speaker model, moving only the weights. As for the second level model, only a few weights are adapted, the other parameters are taken from the upper level models (the corresponding speaker model and the UBM).

5.1.2. Database and Experimentations

The system, currently in development, is evaluated using the Valid database [28] (as the final objective is to build a bimodal system: audio+video). This audio-video Database consists of five recording sessions composed of two sentences (the phonetic content is identical to that of the XM2VTS database 7). Four of these sessions were recorded in variable and uncontrolled environments, the latter in a clean environment. 76 male speakers are used for the experiments.

<table>
<thead>
<tr>
<th>Passwords &amp; Wrong Sentences</th>
<th>GMM</th>
<th>LIA_EBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passwords</td>
<td>1.81</td>
<td>3.09</td>
</tr>
<tr>
<td>Wrong Sentences</td>
<td>2.50</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Table 2: Performance in EER (%) for the GMM and LIA_EBD on the Valid database with different test sets, male only. The experiments are performed by computing 100 client tests, 3275 impostor tests with the same password pronounced by the impostor cohort (the tests are referred to as Passwords) and 3275 tests with the second sentence pronounced by the impostor cohort (referred to as Wrong Sentences)

5.1.3. Comments

The results are promising as the system seems able to exploit both speaker dependent information and password dependent information (the EER decreases from 3.09% to 0.19% when the impostors pronounce a wrong password for the embedded system. The GMM system stays at 1.17% with the wrong sentences). The level of resources (memory and computation) remains very small for the LIA_EBD system. Of course, the presented results are preliminary as the database is very small (76 speakers, two sentences) but only a few system parameters have been optimized and much room remains for improvements.

5.2. Speaker diarization

The design of efficient indexing algorithms to facilitate the retrieval of relevant information is vital to provide easy access to multimedia documents. Acoustic-based information such as speaker turns, the number of speakers, speaker gender, speaker identity, other sounds (music, laughs) as well as speech bandwidth or characteristics (studio quality or telephone speech, clean speech or speech over music) has become mandatory for indexing algorithms. This section is dedicated to speaker-related tasks, also denoted speaker diarization in the NIST-RT evaluation terminology. The speaker diarization task consists of segmenting a conversation involving multiple speakers into homogeneous parts which contain the voice of only one speaker, and grouping together all the segments that correspond to the same speaker. The first part of the process is also called speaker change detection while the second one is known as the clustering process. Generally, no prior information is available regarding the number of speakers involved or their identities. Estimating the number of speakers is one of the main difficulties for the speaker diarization task. An original approach, denoted the integrated approach, in which all the steps involved in speaker diarization are performed simultaneously, was developed by the LIA, using ALIZE/SpkDet software. This work was done in the framework of the ELISA consortium and both classical step-by-step and integrated diarization approaches are proposed in the software, with a lot of functionalities in order to deal with different natures of audio files (conversational recordings, broadcast news or meeting information). This work is detailed in [29]. Table 3 presents a summary of the results obtained between 2002 and 2004 on different types of data, in the framework of the NIST evaluations (mainly on Rich Transcription). More recent results on meeting data were reported in [30][31].

<table>
<thead>
<tr>
<th>Year</th>
<th>Corpus</th>
<th>Teleph.</th>
<th>BN</th>
<th>Meeting head mic.</th>
<th>Meeting table mic.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td></td>
<td>5.7</td>
<td>30.3</td>
<td>34.7</td>
<td>36.9</td>
</tr>
<tr>
<td>spring 2003</td>
<td>X</td>
<td>12.9</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>spring 2004</td>
<td>X</td>
<td></td>
<td>X</td>
<td>22.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: 2002-2004 NIST evaluations results in diarization error rates for various corpora

5.3. Pathological voice assessment

In the medical domain, assessment of the pathological voice quality is a sensitive topic, involving multi-disciplinary domains. This task is usually realized by a jury composed of experts. The work presented in this paragraph proposes to rely on an automatic system for pathological voice assessment, more precisely for dysphonic voices assessment. The proposed method relies on an automatic system derived from the ALIZE/SpkDet automatic speaker recognition technology [32, 33, 34].

Basically, the task consists in labelling a recording or a part of a recording with a label between 0 and 3 (the G parameter of the Hirano’s GRBAS scale[35]). Table 4 presents the general recognition results (given that a strict classification protocol was used, with a clear separation between training and testing set). The obtained results demonstrate the possibilities of this approach, even if the task is very difficult, due to its medical nature, the variability of the data (and the small number of data available) and the difficulty in obtaining a ground truth (done here by an expert jury).

6. Conclusion and future

This paper presents the ALIZE/SpkDet open-source software. The ALIZE project began in 2003 inside the ELISA consortium and under the supervision of LIA laboratory speaker...
The software is dedicated to text-independent speaker recognition but it can also be used for a large range of applications. In addition to the applications presented in this paper, ALIZE/SpkDet software is also used for several other areas, within LIA (face and fingerprint recognition, emotion detection, topic detection for text processing, etc.) or by other users (for example signature/multimodal recognition [36]).

The main part of the development was done by the LIA team (thanks to several French Research Ministry funds) until 2005/2006. Recently, several external (not from the LIA) contributions have been proposed. In order to respond to the requests of the ALIZE/SpkDet developer community, the project has been migrated from a quite closed software engineering platform to a classical open-source open architecture. An important effort will also be dedicated to the documentation.

The main challenge to the ALIZE project is not at the technical level but is to propose a new concept inside the evaluation paradigm: the participation to the international evaluation campaign is viewed as a competition and quite in opposition to the open source idea. Regarding this challenge, it seems that the ALIZE project is a success. In order to emphasize this aspect, we wish to propose inside the ALIZE project some physical evaluation campaigns (as proposed in recent years). The participation to the international evaluation campaigns remains unchanged once the years.

The latest discriminant approaches and channel compensation functionalities were added to the toolkit during the past year leading to state-of-the-art speaker recognition performance. The software is now regularly used by about 20 laboratories in the world, and support is provided to all the developers.

The main part of the development was done by the LIA team (thanks to several French Research Ministry funds) until 2005/2006. Recently, several external (not from the LIA) contributions have been proposed. In order to respond to the requests of the ALIZE/SpkDet developer community, the project has been migrated from a quite closed software engineering platform to a classical open-source open architecture. An important effort will also be dedicated to the documentation.

### References


### Table 4: Pathological voice assessment results. The rows separate the voice depending on the G value given by the jury of experts; the columns represent the G values determined by the system

<table>
<thead>
<tr>
<th>G=0</th>
<th>G=1</th>
<th>G=2</th>
<th>G=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>15</td>
</tr>
</tbody>
</table>


