2

Combining and Comparing Multiple Algorithms for Better Learning and Classification: A Case Study of MARF

Serguei A. Mokhov
Concordia University, Montreal, QC, Canada

1. Introduction

This case study of MARF, an open-source Java-based Modular Audio Recognition Framework, is intended to show the general pattern recognition pipeline design methodology and, more specifically, the supporting interfaces, classes and data structures for machine learning in order to test and compare multiple algorithms and their combinations at the pipeline's stages, including supervised and unsupervised, statistical, etc. learning and classification. This approach is used for a spectrum of recognition tasks, not only applicable to audio, but rather to general pattern recognition for various applications, such as in digital forensic analysis, writer identification, natural language processing (NLP), and others.

2. Chapter overview

First, we present the research problem at hand in Section 3. This is to serve as an example of what researchers can do and choose for their machine learning applications – the types of data structures and the best combinations of available algorithm implementations to suit their needs (or to highlight the need to implement better algorithms if the ones available are not adequate). In MARF, acting as a testbed, the researchers can also test the performance of their own, external algorithms against the ones available. Thus, the overview of the related software engineering aspects and practical considerations are discussed with respect to the machine learning using MARF as a case study with appropriate references to our own and others' related work in Section 4 and Section 5. We discuss to some extent the design and implementation of the data structures and the corresponding interfaces to support learning and comparison of multiple algorithms and approaches in a single framework, and the corresponding implementing system in a consistent environment in Section 6. There we also provide the references to the actual practical implementation of the said data structures within the current framework. We then illustrate some of the concrete results of various MARF applications and discuss them in that perspective in Section 7. We conclude afterwards in Section 8 by outlining some of the advantages and disadvantages of the framework approach and some of the design decisions in Section 8.1 and lay out future research plans in Section 8.2.

3. Problem

The main problem we are addressing is to provide researchers with a tool to test a variety of pattern recognition and NLP algorithms and their combinations for whatever task at hand there is, and then select the best available combination(s) for that final task. The testing should be in a uniform environment to compare and contrast all kinds of algorithms, their parameters, at all stages, and gather metrics such as the precision, run-time, memory usage, recall, f-measure, and others. At the same time, the framework should allow for adding external plug-ins for algorithms written elsewhere as wrappers implementing the framework's API for the same comparative studies.

The system built upon the framework has to have the data structures and interfaces that support such types of experiments in a common, uniform way for comprehensive comparative studies and should allow for scripting of the recognition tasks (for potential batch, distributed, and parallel processing).

These are very broad and general requirements we outlined, and further we describe our approach to them to a various degree using what we call the *Modular Audio Recognition Framework* (MARF). Over the course of years and efforts put into the project, the term *Audio* in the name became a lot less descriptive as the tool grew to be a lot more general and applicable to the other domains than just audio and signal processing, so we will refer to the framework as just *MARF* (while reserving the right to rename it later).

Our philosophy also includes the concept that the tool should be publicly available as an open-source project such that any valuable input and feedback from the community can help everyone involved and make it for the better experimentation platform widely available to all who needs it. Relative simplicity is another aspect that we require the tool to be to be usable by many.

To enable all this, we need to answer the question of "How do we represent what we learn and how do we store it for future use?" What follows is the summary of our take on answering it and the relevant background information.

4. Related work

There are a number of items in the related work; most of them were used as a source to gather the algorithms from to implement within MARF. This includes a variety of classical distance classifiers, such as Euclidean, Chebyshev (a.k.a city-block), Hamming, Mahalanobis, Minkowski, and others, as well as artificial neural networks (ANNs) and all the supporting general mathematics modules found in Abdi (2007); Hamming (1950); Mahalanobis (1936); Russell & Norvig (1995). This also includes the cosine similarity measure as one of the classifiers described in Garcia (2006); Khalifé (2004). Other related work is of course in digital signal processing, digital filters, study of acoustics, digital communication and speech, and the corresponding statistical processing; again for the purpose of gathering of the algorithms for the implementation in a uniform manner in the framework including the ideas presented in Bernsee (1999–2005); Haridas (2006); Haykin (1988); Ifeachor & Jervis (2002); Jurafsky & Martin (2000); O'Shaughnessy (2000); Press (1993); Zwicker & Fastl (1990). These primarily include the design and implementation of the Fast Fourier Transform (FFT) (used for both preprocessing as in low-pass, high-pass, band-pass, etc. filters as well as in feature extraction), Linear Predictive Coding (LPC), Continuous Fraction Expansion (CFE) filters and the corresponding testing applications

implemented by Clement, Mokhov, Nicolacopoulos, Fan & the MARF Research & Development Group (2002–2010); Clement, Mokhov & the MARF Research & Development Group (2002–2010); Mokhov, Fan & the MARF Research & Development Group (2002–2010b; 2005–2010a); Sinclair et al. (2002–2010).
Combining algorithms, an specifically, classifiers is not new, e.g. see Cavalin et al. (2010); Khalifé (2004). We, however, get to combine and chain not only classifiers but algorithms at every stage of the pattern recognition pipeline.
Some of the spectral techniques and statistical techniques are also applicable to the natural language processing that we also implement in some form Jurafsky & Martin (2000); Vaillant et al. (2006); Zipf (1935) where the text is treated as a signal.
Finally, there are open-source speech recognition frameworks, such as CMU Sphinx (see The Sphinx Group at Carnegie Mellon (2007–2010)) that implement a number of algorithms for speech-to-text translation that MARF does not currently implement, but they are quite complex to work with. The advantages of Sphinx is that it is also implemented in Java and is under the same open-source license as MARF, so the latter can integrate the algorithms from Sphinx as external plug-ins. Its disadvantages for the kind of work we are doing are its size and complexity.

5. Our approach and accomplishments

MARF's approach is to define a common set of integrated APIs for the pattern recognition pipeline to allow flexible comparative environment for diverse algorithm implementations for sample loading, preprocessing, feature extraction, and classification. On top of that, the algorithms within each stage can be composed and chained. The conceptual pipeline is shown in Figure 1 and the corresponding UML sequence diagram, shown in Figure 2, details the API invocation and message passing between the core modules, as per Mokhov (2008d); Mokhov et al. (2002–2003); The MARF Research and Development Group (2002–2010).

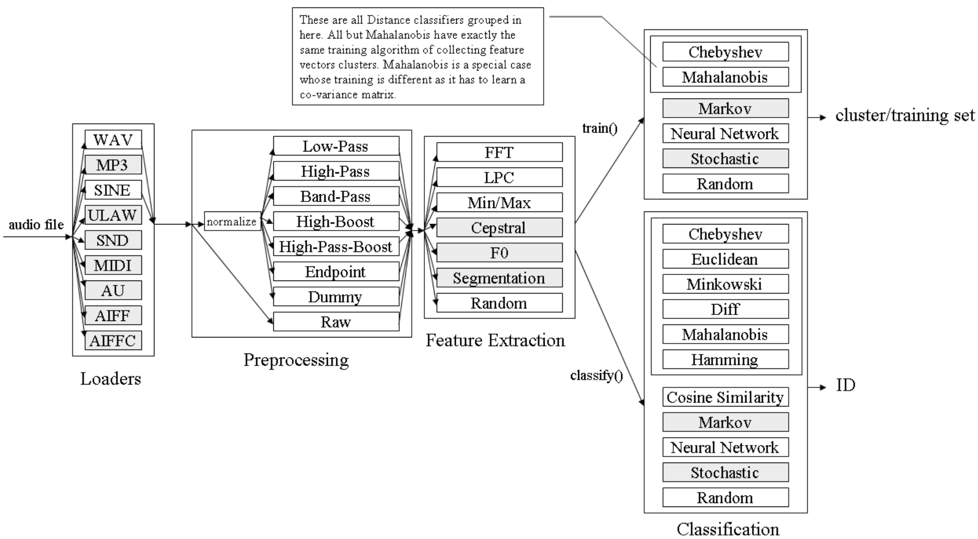


Fig. 1. Classical Pattern Recognition Pipeline of MARF

MARF has been published or is under review and publication with a variety of experimental pattern recognition and software engineering results in multiple venues. The core founding works for this chapter are found in Mokhov (2008a;d; 2010b); Mokhov & Debbabi (2008); Mokhov et al. (2002–2003); The MARF Research and Development Group (2002–2010).
At the beginning, the framework evolved for stand-alone, mostly sequential, applications with limited support for multithreading. Then, the next natural step in its evolution was to make it distributed. Having a distributed MARF (DMARF) still required a lot of manual management, and a proposal was put forward to make it into an autonomic system. A brief overview of the distributed autonomic MARF (DMARF and ADMARF) is given in terms of how the design and practical implementation are accomplished for local and distributed learning and self-management in Mokhov (2006); Mokhov, Huynh & Li (2007); Mokhov et al. (2008); Mokhov & Jayakumar (2008); Mokhov & Vassev (2009a); Vassev & Mokhov (2009; 2010) primarily relying on distributed technologies provided by Java as described in Jini Community (2007); Sun Microsystems, Inc. (2004; 2006); Wollrath & Waldo (1995–2005).
Some scripting aspects of MARF applications are also formally proposed in Mokhov (2008f). Additionally, another frontier of the MARF's use in security is explored in Mokhov (2008e); Mokhov, Huynh, Li & Rassai (2007) as well as the digital forensics aspects that are discussed for various needs of forensic file type analysis, conversion of the MARF's internal data structures as MARFL expressions into the Forensic Lucid language for follow up forensic analysis, self-forensic analysis of MARF, and writer identification of hand-written digitized documents described in Mokhov (2008b); Mokhov & Debbabi (2008); Mokhov et al. (2009); Mokhov & Vassev (2009c).
Furthermore, we have a use case and applicability of MARF's algorithms for various multimedia tasks, e.g. as described in Mokhov (2007b) combined with PureData (see Puckette & PD Community (2007–2010)) as well as in simulation of a solution to the intelligent systems challenge problem Mokhov & Vassev (2009b) and simply various aspects of software engineering associated with the requirements, design, and implementation of the framework outlined in Mokhov (2007a); Mokhov, Miladinova, Ormandjieva, Fang & Amirghahari (2008–2010).
Some MARF example applications, such as text-independent speaker-identification, natural and programming language identification, natural language probabilistic parsing, etc. are released along with MARF as open-source and are discussed in several publications mentioned earlier, specifically in Mokhov (2008–2010c); Mokhov, Sinclair, Clement, Nicolacopoulos & the MARF Research & Development Group (2002–2010); Mokhov & the MARF Research & Development Group (2003–2010a;-), as well as voice-based authentication application of MARF as an utterance engine is in a proprietary VocalVeritas system. The most recent advancements in MARF's applications include the results on identification of the decades and place of origin in the francophone press in the DEFT2010 challenge presented in Forest et al. (2010) with the results described in Mokhov (2010a;b).

6. Methods and tools

To keep the framework flexible and open for comparative uniform studies of algorithms and their external plug-ins we need to define a number of interfaces that the main modules would implement with the corresponding well-documented API as well as what kind of data structures they exchange and populate while using that API. We have to provide the data structures to encapsulate the incoming data for processing as well as the data

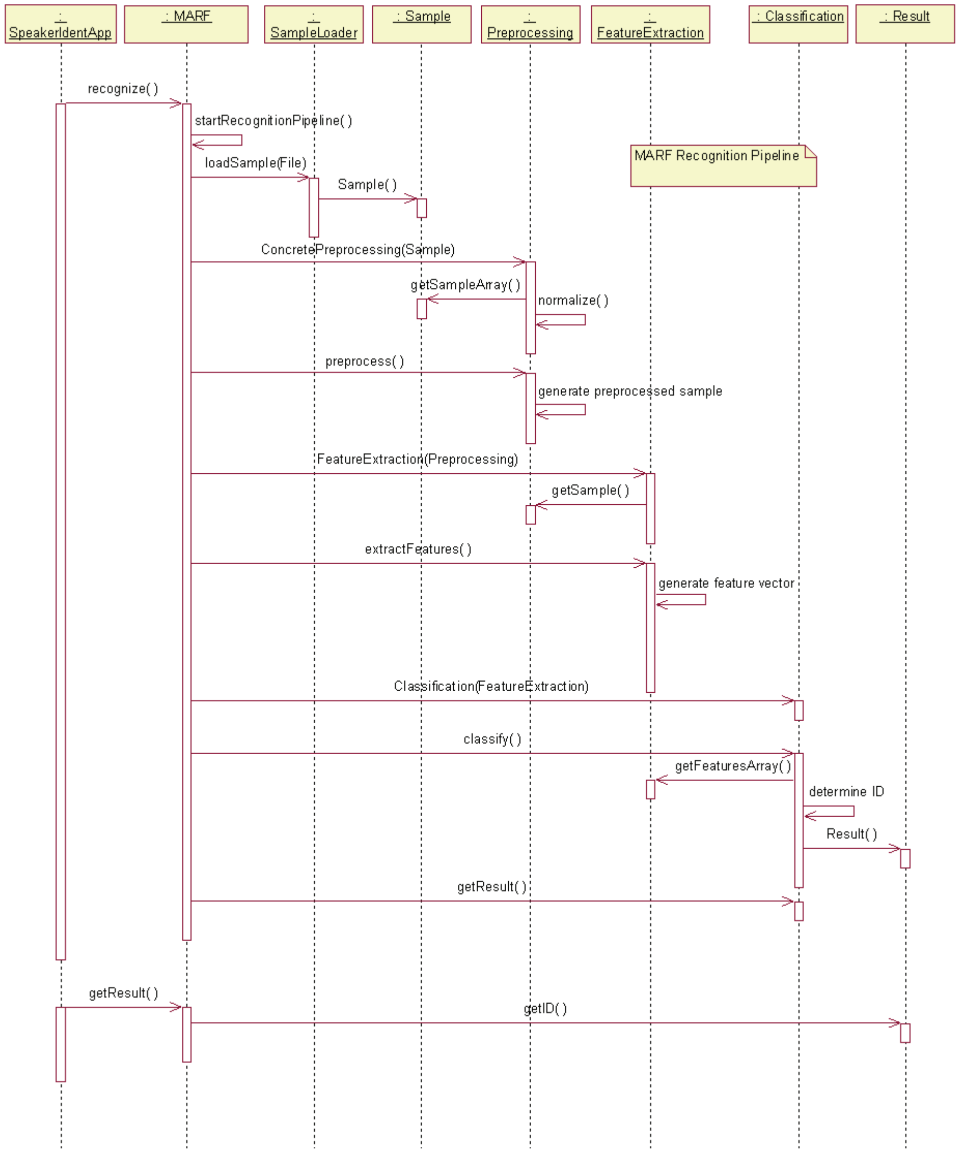


Fig. 2. UML Sequence Diagram of the Classical Pattern Recognition Pipeline of MARF

structures to store the processed data for later retrieval and comparison. In the case of classification, it is necessary also to be able to store more than one classification result, a result set, ordered according to the classification criteria (e.g. sorted in ascending manner for minimal distance or in descending manner for higher probability or similarity). The external applications should be able to pass configuration settings from their own options to the MARF's configuration state as well as collect back the results and aggregate statistics.

While algorithm modules are made fit into the same framework, they all may have arbitrary number of reconfigurable parameters for experiments (e.g. compare the behavior of the same algorithm under different settings) that take some defaults if not explicitly specified. There has to be a generic way of setting those parameters by the applications that are built upon the framework, whose Javadoc's API is detailed here: http://marf.sourceforge.net/api-dev/.

In the rest of the section we describe what we used to achieve the above requirements.

1. We use the Java programming language and the associated set of tools from Sun Microsystems, Inc. (1994–2009) and others as our primary development and run-time environment. This is primarily because it is dynamic, supports reflection (see Green (2001– 2005)), various design patterns and OO programming (Flanagan (1997); Merx & Norman (2007)), exception handling, multithreading, distributed technologies, collections, and other convenient built-in features. We employ Java interfaces for the most major modules to allow for plug-ins.
2. All objects involved in storage are `Serializable`, such that they can be safely stored on disk or transmitted over the network.
3. Many of the data structures are also `Cloneable` to aid copying of the data structure the Java standard way.
4. All major modules in the classical MARF pipeline implement the `IStorageManager` interface, such that they know how to save and reload their state. The default API of `IStorageManager` provides for modules to implement their serialization in a variety of binary and textual formats. Its latest open-source version is at:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/IStorageManager.java?view=markup
5. The `Configuration` object instance is designed to encapsulate the global state of a MARF instance. It can be set by the applications, saved and reloaded or propagated to the distributed nodes. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Configuration.java?view=markup
6. The module parameters class, represented as `ModuleParams`, allows more fine-grained settings for individual algorithms and modules – there can be arbitrary number of the settings in there. Combined with `Configuration` it's the way for applications to pass the specific parameters to the internals of the implementation for diverse experiments. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/ModuleParams.java?view=markup
7. The `Sample` class represents the values either just loaded from an external source (e.g. a file) for preprocessing, or a "massaged" version thereof that was preprocessed already (e.g. had its noise and silence removed, filtered otherwise, and normalized) and is ready for feature extraction. The `Sample` class has a buffer of `Double` values (an array) representing the amplitudes of the sample values being processed at various frequencies and other parameters. It is not important that the input data may be an audio signal, a text, an image, or any kind of binary data – they all can be treated similarly in the spectral approach, so only one way to represent them such that all the modules can understand them. The `Sample` instances are usually of arbitrary length. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/Sample.java?view=markup
8. The `ITrainingSample` interface is very crucial to specify the core storage models for all training samples and training sets. The latter are updated during the training mode of the classifiers and used in read-only manner during the classification stage. The interface also defines what and how to store of the data and how to accumulate the feature vectors that come from the feature extraction modules. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/ITrainingSample.java?view=markup

9. The `TrainingSample` class is the first implementation of the `ITrainingSample` interface. It maintains the ID of the subject that training sample data corresponds to, the training data vector itself (usually either a mean or median cluster or a single feature vector), and a list of files (or entries alike) the training was performed on (this list is optionally used by the classification modules to avoid double-training on the same sample). Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/TrainingSample.java?view=markup
10. The `Cluster` is a `TrainingSample` with a mean cluster data embedded and counted how many feature vectors were particularly trained on. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/Cluster.java?view=markup
11. The `TrainingSet` class encapsulates a collection of object instances implementing the `ITrainingSample` interface and whether they are simply `TrainingSamples`, `Clusters`, or `FeatureSets`. It also caries the information about which preprocessing and feature extraction methods were used to disambiguate the sets. Most commonly, the serialized instances of this class are preserved during the training sessions and used during the classification sessions. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/TrainingSet.java?view=markup
12. The `FeatureSet` class instance is a `Cluster` that allows maintaining individual feature vectors instead of just a compressed (mean or median) clusters thereof. It allows for the most flexibility and retains the most training information available at the cost of extra storage and look up requirements. The flexibility allows to compute the mean and median vectors and cache them dynamically if the feature set was not altered increasing performance. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/FeatureSet.java?view=markup
13. An instance of the `Result` data structure encapsulates the classification ID (usually supplied during training), the outcome for that result, and a particular optional description if required (e.g. human-readable interpretation of the ID). The outcome may mean a number of things depending on the classifier used: it is a scalar `Double` value that can represent the distance from the subject, the similarity to the subject, or probability of this result. These meanings are employed by the particular classifiers when returning the “best” and “second best”, etc. results or sort them from the “best” to the “worst” whatever these qualifiers mean. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/Result.java?view=markup
14. The `ResultSet` class corresponds to the collection of `Results`, that can be sorted according to each classifier’s requirements. It provides the basic API to get minima, maxima (both first, and second), as well as average and random and the entire collection of the results. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/ResultSet.java?view=markup
15. The `IDatabase` interface is there to be used by applications to maintain their instances of database abstractions to maintain statistics they need, such as precision of recognition, etc. generally following the Builder design pattern (see Freeman et al. (2004); Gamma et al. (1995); Larman (2006)). Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/IDatabase.java?view=markup
16. The `Database` class instance is the most generic implementation of the `IDatabase` interface in case applications decide to use it. The applications such as `SpeakerIdentApp`, `WriterIdentApp`, `FileTypeIdentApp`, `DEFT2010App` and others have their corresponding subclasses of this class. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Storage/Database.java?view=markup

17. The `StatisticalObject` class is a generic record about frequency of occurrences and potentially a rank of any statistical value. In MARF, typically it is the basis for various NLP-related observations. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Stats/StatisticalObject.java?view=markup
18. The `WordStats` class is a `StatisticalObject` that is more suitable for text analysis and extends it with the lexeme being observed. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Stats/WordStats.java?view=markup
19. The `Observation` class is a refinement of `WordStats` to augment it with prior and posterior probabilities as well as the fact it has been "seen" or not yet. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Stats/Observation.java?view=markup
20. The `Ngram` instance is an `Observation` of an occurrence of an n-ngram usually in the natural language text with $n = 1, 2, 3, \ldots$ characters or lexeme elements that follow each other. Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Stats/Ngram.java?view=markup
21. The `ProbabilityTable` class instance builds matrices of n-grams and their computed or counted probabilities for training and classification (e.g. in `LangIdentApp`). Details:
http://marf.cvs.sf.net/viewvc/marf/marf/src/marf/Stats/ProbabilityTable.java?view=markup

7. Results

We applied the MARF approach to a variety experiments, that gave us equally a variety of results. The approaches tried refer to text independent-speaker identification using median and mean clusters, gender identification, age group, spoken accent, and biometrics alike. On the other hand, other experiments involved writer identification from scanned hand-written documents, forensic file type analysis of file systems, an intelligent systems challenge, natural language identification, identification of decades in French corpora as well as place of origin of publication (such as Quebec vs. France or the particular journal).

All these experiments yielded top, intermediate, and worst configurations for each task given the set of available algorithms implemented at the time. Here we recite some of the results with their configurations. This is a small fraction of the experiments conducted and results recorded as a normal session is about ≈ 1500+ configurations.

1. Text-independent speaker (Mokhov (2008a;c); Mokhov et al. (2002–2003)), including gender, and spoken accent identification using mean vs. median clustering experimental (Mokhov (2008a;d)) results are illustrated in Table 1, Table 2, Table 3, Table 4, Table 5, and Table 6. These are primarily results with the top precision. The point these serve to illustrate is that the top configurations of algorithms are distinct depending on (a) the recognition task ("who" vs. "spoken accent" vs. "gender") and (b) type of clustering performed. For instance, by using the mean clustering the configuration that removes silence gaps from the sample, uses the band-stop FFT filter, and uses the aggregation of the FFT and LPC features in one feature vector and the cosine similarity measure as the classifier yielded the top result in Table 1. However, an equivalent experiment in Table 2 with median clusters yielded band-stop FFT filter with FFT feature extractor and cosine similarity classifier as a top configuration; and the configuration that was the top for the mean was no longer that accurate. The individual modules used in the pipeline were all at their default settings (see Mokhov (2008d)). The meanings of the options are also described in Mokhov (2008d; 2010b); The MARF

Rank #	Configuration	$GOOD_{1st}$	BAD_{1st}	$Precision_{1st}$,%	$GOOD_{2nd}$	BAD_{2nd}	$Precision_{2nd}$,%
1	-silence -bandstop -aggr -cos	29	3	90.62	30	2	93.75
1	-silence -bandstop -fft -cos	29	3	90.62	30	2	93.75
1	-bandstop -fft -cos	28	4	87.50	29	3	90.62
2	-silence -noise -bandstop -fft -cos	28	4	87.50	30	2	93.75
2	-silence -low -aggr -cos	28	4	87.50	30	2	93.75
2	-silence -noise -norm -aggr -cos	28	4	87.50	30	2	93.75
2	-silence -low -fft -cos	28	4	87.50	30	2	93.75
2	-silence -noise -norm -fft -cos	28	4	87.50	30	2	93.75
2	-silence -noise -low -aggr -cos	28	4	87.50	30	2	93.75
2	-silence -noise -low -fft -cos	28	4	87.50	30	2	93.75
2	-bandstop -aggr -cos	28	4	87.50	29	3	90.62
2	-norm -fft -cos	28	4	87.50	29	3	90.62
2	-silence -raw -aggr -cos	28	4	87.50	30	2	93.75
2	-silence -noise -raw -aggr -cos	28	4	87.50	30	2	93.75
2	-norm -aggr -cos	28	4	87.50	30	2	93.75
2	-silence -noise -bandstop -aggr -cos	28	4	87.50	30	2	93.75
3	-silence -norm -fft -cos	27	5	84.38	30	2	93.75
3	-silence -norm -aggr -cos	27	5	84.38	30	2	93.75
3	-low -fft -cos	27	5	84.38	28	4	87.50
3	-noise -bandstop -aggr -cos	27	5	84.38	29	3	90.62
3	-silence -raw -fft -cos	27	5	84.38	29	3	90.62
3	-noise -raw -aggr -cos	27	5	84.38	30	2	93.75
3	-silence -noise -raw -fft -cos	27	5	84.38	29	3	90.62
3	-noise -low -fft -cos	27	5	84.38	28	4	87.50
3	-raw -fft -cos	27	5	84.38	29	3	90.62
3	-noise -bandstop -fft -cos	27	5	84.38	29	3	90.62
3	-low -aggr -cos	27	5	84.38	28	4	87.50
3	-noise -raw -fft -cos	27	5	84.38	29	3	90.62
3	-noise -norm -fft -cos	27	5	84.38	28	4	87.50
3	-noise -norm -aggr -cos	27	5	84.38	28	4	87.50
3	-noise -low -aggr -cos	27	5	84.38	28	4	87.50
4	-noise -raw -lpc -cos	26	6	81.25	28	4	87.50
4	-silence -raw -lpc -cos	26	6	81.25	28	4	87.50
4	-silence -noise -raw -lpc -cos	26	6	81.25	28	4	87.50
4	-raw -lpc -cos	26	6	81.25	28	4	87.50
4	-norm -lpc -cos	26	6	81.25	28	4	87.50
5	-endp -lpc -cheb	25	7	78.12	26	6	81.25
6	-silence -bandstop -fft -eucl	24	8	75.00	26	6	81.25
6	-bandstop -lpc -eucl	24	8	75.00	28	4	87.50
6	-silence -norm -fft -eucl	24	8	75.00	26	6	81.25
6	-silence -bandstop -fft -diff	24	8	75.00	26	6	81.25
6	-silence -norm -aggr -eucl	24	8	75.00	26	6	81.25
6	-raw -fft -eucl	24	8	75.00	26	6	81.25
6	-noise -raw -aggr -eucl	24	8	75.00	26	6	81.25
6	-silence -bandstop -aggr -eucl	24	8	75.00	26	6	81.25
6	-bandstop -aggr -cheb	24	8	75.00	26	6	81.25
6	-noise -raw -fft -eucl	24	8	75.00	26	6	81.25
6	-silence -raw -fft -eucl	24	8	75.00	26	6	81.25
6	-silence -bandstop -aggr -diff	24	8	75.00	26	6	81.25
6	-silence -noise -raw -aggr -eucl	24	8	75.00	26	6	81.25

Table 1. Top Most Accurate Configurations for Speaker Identification, 1st and 2nd Guesses, Mean Clustering (Mokhov (2008d))

Research and Development Group (2002–2010). We also illustrate the "2nd guess" statistics – often what happens is that if we are mistaken in our first guess, the second one is usually the right one. It may not be obvious how to exploit it, but we provide the statistics to show if the hypothesis is true or not.

While the options listed of the MARF application (`SpeakerIdentApp`, see Mokhov, Sinclair, Clement, Nicolacopoulos & the MARF Research & Development Group (2002–2010)) are described at length in the cited works, here we briefly summarize their meaning for the unaware reader: `-silence` and `-noise` tell to remove the silence and noise components of a sample; `-band`, `-bandstop`, `-high` and `-low` correspond to the band-pass, band-stop, high-pass and low-pass FFT filters; `-norm` means normalization; `-endp` corresponds to endpointing; `-raw` does a pass-through (no-op) preprocessing;

Rank #	Configuration	$GOOD_{1st}$	BAD_{1st}	$Precision_{1st}$,%	$GOOD_{2nd}$	BAD_{2nd}	$Precision_{2nd}$,%
1	-bandstop -fft -cos	29	3	90.62	30	2	93.75
1	-bandstop -aggr -cos	29	3	90.62	30	2	93.75
2	-silence -bandstop -aggr -cos	28	4	87.5	30	2	93.75
2	-silence -bandstop -fft -cos	28	4	87.5	30	2	93.75
2	-low -fft -cos	28	4	87.5	29	3	90.62
2	-noise -bandstop -aggr -cos	28	4	87.5	29	3	90.62
2	-silence -raw -fft -cos	28	4	87.5	30	2	93.75
2	-noise -raw -aggr -cos	28	4	87.5	30	2	93.75
2	-silence -noise -raw -fft -cos	28	4	87.5	30	2	93.75
2	-noise -low -fft -cos	28	4	87.5	29	3	90.62
2	-raw -fft -cos	28	4	87.5	30	2	93.75
2	-noise -bandstop -fft -cos	28	4	87.5	29	3	90.62
2	-norm -fft -cos	28	4	87.5	30	2	93.75
2	-noise -raw -fft -cos	28	4	87.5	30	2	93.75
2	-noise -norm -fft -cos	28	4	87.5	29	3	90.62
2	-noise -low -aggr -cos	28	4	87.5	29	3	90.62
2	-norm -aggr -cos	28	4	87.5	30	2	93.75
3	-silence -norm -fft -cos	27	5	84.38	29	3	90.62
3	-silence -low -aggr -cos	27	5	84.38	30	2	93.75
3	-silence -noise -norm -aggr -cos	27	5	84.38	30	2	93.75
3	-silence -norm -aggr -cos	27	5	84.38	29	3	90.62
3	-silence -low -fft -cos	27	5	84.38	30	2	93.75
3	-silence -noise -norm -fft -cos	27	5	84.38	30	2	93.75
3	-silence -noise -low -aggr -cos	27	5	84.38	30	2	93.75
3	-silence -noise -low -fft -cos	27	5	84.38	30	2	93.75
3	-raw -aggr -cos	27	5	84.38	30	2	93.75
3	-low -aggr -cos	27	5	84.38	29	3	90.62
3	-silence -raw -aggr -cos	27	5	84.38	30	2	93.75
3	-silence -noise -raw -aggr -cos	27	5	84.38	30	2	93.75
3	-noise -norm -aggr -cos	27	5	84.38	29	3	90.62
4	-silence -noise -bandstop -fft -cos	26	6	81.25	30	2	93.75
4	-bandstop -lpc -diff	26	6	81.25	31	1	96.88
4	-bandstop -lpc -cheb	26	6	81.25	31	1	96.88
4	-silence -noise -bandstop -aggr -cos	26	6	81.25	30	2	93.75
5	-bandstop -lpc -eucl	25	7	78.12	31	1	96.88
5	-noise -raw -lpc -cos	25	7	78.12	26	6	81.25
5	-bandstop -lpc -cos	25	7	78.12	29	3	90.62
5	-silence -raw -lpc -cos	25	7	78.12	26	6	81.25
5	-silence -noise -raw -lpc -cos	25	7	78.12	26	6	81.25
5	-raw -lpc -cos	25	7	78.12	26	6	81.25
5	-norm -lpc -cos	25	7	78.12	26	6	81.25
6	-silence -norm -fft -eucl	24	8	75	26	6	81.25
6	-bandstop -fft -cheb	24	8	75	26	6	81.25
6	-silence -norm -aggr -eucl	24	8	75	26	6	81.25
6	-endp -lpc -cheb	24	8	75	27	5	84.38
6	-bandstop -aggr -cheb	24	8	75	26	6	81.25
6	-bandstop -fft -diff	24	8	75	26	6	81.25
6	-bandstop -aggr -diff	24	8	75	26	6	81.25
6	-bandstop -lpc -mink	24	8	75	30	2	93.75
7	-silence -bandstop -fft -eucl	23	9	71.88	26	6	81.25
7	-silence -bandstop -aggr -cheb	23	9	71.88	26	6	81.25
7	-bandstop -fft -eucl	23	9	71.88	26	6	81.25
7	-silence -bandstop -aggr -eucl	23	9	71.88	26	6	81.25
7	-silence -endp -lpc -cheb	23	9	71.88	25	7	78.12
7	-endp -lpc -eucl	23	9	71.88	26	6	81.25

Table 2. Top Most Accurate Configurations for Speaker Identification, 1^{st} and 2^{nd} Guesses, Median Clustering (Mokhov (2008d))

`-fft`, `-lpc`, and `-aggr` correspond to the FFT-based, LPC-based, or aggregation of the two feature extractors; `-cos`, `-eucl`, `-cheb`, `-hamming`, `-mink`, and `–diff` correspond to the classifiers, such as cosine similarity measure, Euclidean, Chebyshev, Hamming, Minkowski, and diff distances respectively.

2. In Mokhov & Debbabi (2008), an experiment was conducted to use a MARF-based `FileTypeIdentApp` for bulk forensic analysis of file types using signal processing techniques as opposed to the Unix `file` utility (see Darwin et al. (1973–2007;-)). That experiment was a “cross product” of:

Rank #	Configuration	$GOOD_{1st}$	BAD_{1st}	$Precision_{1st}$,%	$GOOD_{2nd}$	BAD_{2nd}	$Precision_{2nd}$,%
1	-silence -endp -lpc -cheb	24	8	75	26	6	81.25
2	-bandstop -fft -cos	23	9	71.88	27	5	84.38
2	-low -aggr -cos	23	9	71.88	26	6	81.25
2	-noise -norm -aggr -cos	23	9	71.88	26	6	81.25
2	-noise -low -aggr -cos	23	9	71.88	26	6	81.25
3	-noise -bandstop -aggr -cos	22	10	68.75	27	5	84.38
3	-noise -low -fft -cos	22	10	68.75	26	6	81.25
3	-noise -bandstop -fft -cos	22	10	68.75	27	5	84.38
3	-norm -aggr -cos	22	10	68.75	26	6	81.25
4	-endp -lpc -cheb	21	11	65.62	24	8	75
4	-silence -noise -low -aggr -cos	21	11	65.62	25	7	78.12
4	-low -fft -cos	21	11	65.62	27	5	84.38
4	-noise -norm -fft -cos	21	11	65.62	27	5	84.38
5	-silence -bandstop -aggr -cos	20	12	62.5	25	7	78.12
5	-silence -low -aggr -cos	20	12	62.5	25	7	78.12
5	-silence -noise -norm -aggr -cos	20	12	62.5	25	7	78.12
5	-silence -bandstop -fft -cos	20	12	62.5	25	7	78.12
5	-silence -low -fft -cos	20	12	62.5	25	7	78.12
5	-silence -noise -norm -fft -cos	20	12	62.5	25	7	78.12
5	-silence -noise -low -fft -cos	20	12	62.5	25	7	78.12
5	-endp -lpc -diff	20	12	62.5	24	8	75
5	-norm -fft -cos	20	12	62.5	26	6	81.25
5	-silence -endp -lpc -eucl	20	12	62.5	23	9	71.88
5	-noise -band -lpc -cos	20	12	62.5	26	6	81.25
5	-silence -endp -lpc -diff	20	12	62.5	26	6	81.25
6	-silence -noise -bandstop -fft -cos	19	13	59.38	25	7	78.12
6	-noise -band -fft -eucl	19	13	59.38	23	9	71.88
6	-silence -norm -fft -cos	19	13	59.38	27	5	84.38
6	-silence -norm -aggr -cos	19	13	59.38	27	5	84.38
6	-silence -raw -fft -cos	19	13	59.38	27	5	84.38
6	-silence -noise -band -aggr -mink	19	13	59.38	25	7	78.12
6	-silence -noise -band -fft -mink	19	13	59.38	25	7	78.12
6	-silence -noise -raw -fft -cos	19	13	59.38	27	5	84.38
6	-raw -fft -cos	19	13	59.38	27	5	84.38
6	-silence -noise -bandstop -fft -cheb	19	13	59.38	24	8	75
6	-noise -raw -fft -cos	19	13	59.38	27	5	84.38
6	-noise -endp -lpc -cos	19	13	59.38	25	7	78.12
6	-silence -noise -bandstop -aggr -cos	19	13	59.38	25	7	78.12
7	-silence -noise -bandstop -aggr -cheb	16	12	57.14	20	8	71.43
8	-silence -noise -bandstop -fft -diff	18	14	56.25	25	7	78.12
8	-noise -high -aggr -cos	18	14	56.25	20	12	62.5
8	-silence -endp -lpc -cos	18	14	56.25	23	9	71.88
8	-silence -noise -low -lpc -hamming	18	14	56.25	25	7	78.12
8	-silence -noise -low -aggr -cheb	18	14	56.25	23	9	71.88
8	-silence -noise -endp -lpc -cos	18	14	56.25	25	7	78.12
8	-silence -noise -low -fft -diff	18	14	56.25	22	10	68.75
8	-raw -aggr -cos	18	14	56.25	28	4	87.5
8	-noise -bandstop -fft -diff	18	14	56.25	24	8	75
8	-noise -band -lpc -cheb	18	14	56.25	27	5	84.38
8	-silence -endp -lpc -hamming	18	14	56.25	24	8	75
8	-low -aggr -diff	18	14	56.25	24	8	75
8	-noise -band -fft -cos	18	14	56.25	22	10	68.75
8	-silence -noise -low -aggr -diff	18	14	56.25	23	9	71.88
8	-noise -band -fft -cheb	18	14	56.25	22	10	68.75
8	-silence -band -lpc -cheb	18	14	56.25	21	11	65.62
8	-silence -noise -low -fft -cheb	18	14	56.25	23	9	71.88
8	-noise -bandstop -aggr -cheb	18	14	56.25	25	7	78.12
8	-noise -bandstop -fft -cheb	18	14	56.25	24	8	75
8	-silence -noise -bandstop -aggr -diff	18	14	56.25	25	7	78.12
9	-noise -high -fft -eucl	17	15	53.12	22	10	68.75
9	-noise -high -aggr -eucl	17	15	53.12	20	12	62.5

Table 3. Top Most Accurate Configurations for Spoken Accent Identification, 1st and 2nd Guesses, Mean Clustering (Mokhov (2008d))

- 3 loaders
- strings and *n*-grams (4)
- noise and silence removal (4)
- 13 preprocessing modules
- 5 feature extractors
- 9 classifiers

Run #	Configuration	$GOOD_{1st}$	BAD_{1st}	$Precision_{1st}$,%	$GOOD_{2nd}$	BAD_{2nd}	$Precision_{2nd}$,%
1	-noise -raw -aggr -cos	23	9	71.88	25	7	78.12
1	-silence -noise -raw -aggr -cos	23	9	71.88	25	7	78.12
2	-raw -aggr -cos	22	10	68.75	25	7	78.12
2	-silence -raw -fft -cos	22	10	68.75	25	7	78.12
2	-silence -noise -raw -fft -cos	22	10	68.75	25	7	78.12
2	-raw -fft -cos	22	10	68.75	25	7	78.12
2	-silence -raw -aggr -cos	22	10	68.75	25	7	78.12
2	-noise -raw -fft -cos	22	10	68.75	25	7	78.12
3	-noise -low -aggr -eucl	21	11	65.62	28	4	87.5
3	-band -aggr -cos	21	11	65.62	25	7	78.12
3	-noise -endp -fft -eucl	21	11	65.62	28	4	87.5
3	-low -aggr -cos	21	11	65.62	26	6	81.25
3	-noise -low -fft -eucl	21	11	65.62	28	4	87.5
3	-noise -norm -aggr -cos	21	11	65.62	26	6	81.25
3	-noise -low -aggr -cos	21	11	65.62	27	5	84.38
4	-silence -low -fft -eucl	20	12	62.5	27	5	84.38
4	-silence -noise -bandstop -fft -cos	20	12	62.5	25	7	78.12
4	-silence -noise -bandstop -fft -diff	20	12	62.5	26	6	81.25
4	-silence -norm -fft -eucl	20	12	62.5	27	5	84.38
4	-silence -bandstop -aggr -cos	20	12	62.5	25	7	78.12
4	-silence -bandstop -fft -cos	20	12	62.5	25	7	78.12
4	-silence -noise -norm -fft -eucl	20	12	62.5	27	5	84.38
4	-silence -bandstop -aggr -cheb	20	12	62.5	28	4	87.5
4	-silence -norm -aggr -eucl	20	12	62.5	27	5	84.38
4	-noise -bandstop -fft -eucl	20	12	62.5	27	5	84.38
4	-silence -norm -fft -diff	20	12	62.5	24	8	75
4	-bandstop -fft -eucl	20	12	62.5	27	5	84.38
4	-noise -bandstop -fft -diff	20	12	62.5	24	8	75
4	-silence -low -aggr -eucl	20	12	62.5	27	5	84.38
4	-silence -bandstop -aggr -diff	20	12	62.5	28	4	87.5
4	-silence -noise -bandstop -fft -cheb	20	12	62.5	26	6	81.25
4	-silence -norm -fft -cheb	20	12	62.5	24	8	75
4	-norm -aggr -cos	20	12	62.5	26	6	81.25
4	-silence -noise -bandstop -aggr -cos	20	12	62.5	25	7	78.12
4	-silence -noise -bandstop -aggr -diff	20	12	62.5	26	6	81.25
4	-noise -bandstop -fft -cheb	20	12	62.5	24	8	75
5	-silence -bandstop -fft -eucl	19	13	59.38	28	4	87.5
5	-bandstop -fft -cos	19	13	59.38	26	6	81.25
5	-silence -norm -fft -cos	19	13	59.38	26	6	81.25
5	-silence -low -aggr -cos	19	13	59.38	25	7	78.12
5	-silence -noise -low -fft -eucl	19	13	59.38	27	5	84.38
5	-silence -norm -aggr -cos	19	13	59.38	26	6	81.25
5	-silence -bandstop -fft -diff	19	13	59.38	27	5	84.38
5	-silence -low -fft -cos	19	13	59.38	25	7	78.12
5	-silence -low -fft -diff	19	13	59.38	23	9	71.88
5	-silence -noise -low -lpc -hamming	19	13	59.38	23	9	71.88
5	-endp -lpc -cheb	19	13	59.38	23	9	71.88
5	-noise -bandstop -aggr -mink	19	13	59.38	24	8	75
5	-silence -noise -band -fft -cheb	19	13	59.38	25	7	78.12
5	-noise -bandstop -aggr -eucl	19	13	59.38	27	5	84.38
5	-silence -noise -norm -fft -cos	19	13	59.38	25	7	78.12
5	-silence -noise -low -aggr -cos	19	13	59.38	25	7	78.12
5	-silence -noise -low -aggr -cheb	19	13	59.38	25	7	78.12
5	-silence -noise -endp -lpc -cos	19	13	59.38	26	6	81.25
5	-noise -raw -aggr -mink	19	13	59.38	24	8	75
5	-silence -low -aggr -cheb	19	13	59.38	23	9	71.88
5	-low -aggr -eucl	19	13	59.38	27	5	84.38
5	-low -fft -cos	19	13	59.38	26	6	81.25
5	-silence -noise -low -fft -cos	19	13	59.38	25	7	78.12
5	-noise -bandstop -aggr -cos	19	13	59.38	21	11	65.62
5	-silence -noise -low -fft -diff	19	13	59.38	25	7	78.12
5	-silence -noise -norm -fft -diff	19	13	59.38	23	9	71.88
5	-raw -aggr -mink	19	13	59.38	23	9	71.88
5	-silence -norm -aggr -diff	19	13	59.38	24	8	75
5	-silence -noise -endp -lpc -cheb	19	13	59.38	26	6	81.25
5	-silence -bandstop -aggr -eucl	19	13	59.38	26	6	81.25
5	-bandstop -aggr -cheb	19	13	59.38	26	6	81.25

Table 4. Top Most Accurate Configurations for Spoken Accent Identification, 1^{st} and 2^{nd} Guesses, Median Clustering (Mokhov (2008d))

Rank #	Configuration	$GOOD_{1st}$	BAD_{1st}	$Precision_{1st}, \%$	$GOOD_{2nd}$	BAD_{2nd}	$Precision_{2nd}, \%$
1	-noise -high -aggr -mink	26	6	81.25	32	0	100
1	-silence -noise -band -aggr -cheb	26	6	81.25	32	0	100
1	-silence -noise -band -lpc -cos	26	6	81.25	31	1	96.88
1	-silence -noise -band -fft -cheb	26	6	81.25	32	0	100
1	-noise -bandstop -fft -diff	26	6	81.25	32	0	100
1	-noise -bandstop -fft -cheb	26	6	81.25	32	0	100
2	-silence -band -lpc -cos	25	7	78.12	31	1	96.88
2	-silence -noise -bandstop -fft -diff	25	7	78.12	32	0	100
2	-noise -endp -lpc -eucl	25	7	78.12	31	1	96.88
2	-silence -noise -band -aggr -eucl	25	7	78.12	32	0	100
2	-silence -noise -endp -lpc -cheb	25	7	78.12	32	0	100
2	-noise -endp -lpc -diff	25	7	78.12	32	0	100
2	-silence -noise -band -fft -eucl	25	7	78.12	32	0	100
2	-silence -noise -band -aggr -diff	25	7	78.12	32	0	100
2	-silence -noise -bandstop -fft -cheb	25	7	78.12	32	0	100
2	-silence -noise -band -fft -diff	25	7	78.12	32	0	100
2	-noise -bandstop -aggr -cheb	25	7	78.12	32	0	100
3	-noise -band -aggr -cheb	24	8	75	32	0	100
3	-noise -high -fft -eucl	24	8	75	31	1	96.88
3	-noise -high -lpc -cos	24	8	75	30	2	93.75
3	-silence -low -fft -diff	24	8	75	32	0	100
3	-silence -noise -high -lpc -diff	24	8	75	30	2	93.75
3	-silence -noise -low -aggr -cheb	24	8	75	32	0	100
3	-silence -noise -endp -lpc -cos	24	8	75	31	1	96.88
3	-silence -noise -low -fft -diff	24	8	75	32	0	100
3	-silence -noise -norm -fft -diff	24	8	75	32	0	100
3	-silence -noise -norm -aggr -cheb	24	8	75	32	0	100
3	-silence -noise -bandstop -aggr -cheb	24	8	75	32	0	100
3	-silence -noise -endp -lpc -eucl	24	8	75	31	1	96.88
3	-silence -noise -low -aggr -diff	24	8	75	32	0	100
3	-silence -noise -norm -aggr -diff	24	8	75	32	0	100
3	-noise -endp -lpc -cos	24	8	75	31	1	96.88
3	-silence -noise -low -fft -cheb	24	8	75	32	0	100
3	-noise -endp -lpc -hamming	24	8	75	31	1	96.88
3	-silence -noise -bandstop -aggr -diff	24	8	75	32	0	100
3	-noise -endp -lpc -cheb	24	8	75	32	0	100
4	-low -lpc -cheb	23	9	71.88	32	0	100
4	-noise -norm -lpc -cheb	23	9	71.88	32	0	100
4	-noise -low -lpc -cheb	23	9	71.88	32	0	100
4	-endp -lpc -cheb	23	9	71.88	31	1	96.88
4	-noise -band -fft -diff	23	9	71.88	32	0	100
4	-low -lpc -mink	23	9	71.88	31	1	96.88
4	-low -lpc -eucl	23	9	71.88	31	1	96.88
4	-noise -norm -aggr -cheb	23	9	71.88	32	0	100
4	-noise -norm -lpc -mink	23	9	71.88	31	1	96.88
4	-silence -high -lpc -cos	23	9	71.88	32	0	100
4	-noise -low -lpc -mink	23	9	71.88	32	0	100
4	-noise -norm -lpc -eucl	23	9	71.88	31	1	96.88
4	-noise -low -lpc -eucl	23	9	71.88	32	0	100
4	-silence -low -lpc -cheb	23	9	71.88	31	1	96.88
4	-noise -band -lpc -hamming	23	9	71.88	30	2	93.75
4	-noise -band -aggr -diff	23	9	71.88	32	0	100
4	-silence -noise -raw -aggr -cheb	23	9	71.88	32	0	100
4	-endp -lpc -eucl	23	9	71.88	29	3	90.62
4	-low -lpc -diff	23	9	71.88	32	0	100
4	-noise -low -fft -cheb	23	9	71.88	32	0	100
4	-silence -noise -norm -lpc -cheb	23	9	71.88	31	1	96.88
4	-noise -norm -lpc -diff	23	9	71.88	32	0	100
4	-noise -low -lpc -diff	23	9	71.88	32	0	100
4	-endp -lpc -diff	23	9	71.88	31	1	96.88
4	-noise -high -lpc -mink	23	9	71.88	29	3	90.62
4	-noise -high -fft -cheb	23	9	71.88	29	3	90.62
4	-silence -low -fft -cheb	23	9	71.88	32	0	100
4	-silence -noise -high -lpc -cheb	23	9	71.88	30	2	93.75
4	-noise -norm -aggr -diff	23	9	71.88	32	0	100
4	-noise -band -lpc -cos	23	9	71.88	30	2	93.75

Table 5. Top Most Accurate Configurations for Gender Identification, 1st and 2nd Guesses, Mean Clustering (Mokhov (2008d))

Run #	Configuration	$GOOD_{1st}$	BAD_{1st}	$Precision_{1st}$,%	$GOOD_{2nd}$	BAD_{2nd}	$Precision_{2nd}$,%
1	-silence -noise -band -lpc -cos	26	6	81.25	30	2	93.75
1	-silence -noise -endp -lpc -eucl	26	6	81.25	31	1	96.88
2	-silence -band -lpc -cos	25	7	78.12	31	1	96.88
2	-silence -noise -band -aggr -cheb	25	7	78.12	32	0	100
2	-silence -band -lpc -mink	25	7	78.12	32	0	100
2	-endp -lpc -cheb	25	7	78.12	31	1	96.88
2	-silence -noise -band -fft -cheb	25	7	78.12	32	0	100
2	-noise -endp -lpc -eucl	25	7	78.12	31	1	96.88
2	-silence -noise -endp -lpc -cheb	25	7	78.12	32	0	100
2	-silence -noise -band -aggr -diff	25	7	78.12	32	0	100
2	-silence -noise -bandstop -aggr -cheb	25	7	78.12	32	0	100
2	-silence -noise -bandstop -fft -cheb	25	7	78.12	32	0	100
2	-silence -noise -band -fft -diff	25	7	78.12	32	0	100
2	-silence -noise -bandstop -aggr -diff	25	7	78.12	32	0	100
3	-noise -high -aggr -mink	24	8	75	31	1	96.88
3	-low -lpc -cheb	24	8	75	31	1	96.88
3	-silence -noise -bandstop -fft -diff	24	8	75	32	0	100
3	-noise -high -aggr -eucl	24	8	75	30	2	93.75
3	-noise -high -lpc -cos	24	8	75	30	2	93.75
3	-noise -norm -lpc -cheb	24	8	75	31	1	96.88
3	-noise -low -lpc -cheb	24	8	75	32	0	100
3	-noise -bandstop -aggr -eucl	24	8	75	32	0	100
3	-silence -noise -endp -lpc -cos	24	8	75	31	1	96.88
3	-silence -noise -band -lpc -diff	24	8	75	32	0	100
3	-low -lpc -mink	24	8	75	30	2	93.75
3	-low -lpc -eucl	24	8	75	30	2	93.75
3	-noise -norm -lpc -mink	24	8	75	30	2	93.75
3	-noise -low -lpc -mink	24	8	75	30	2	93.75
3	-silence -noise -band -aggr -eucl	24	8	75	32	0	100
3	-noise -norm -lpc -eucl	24	8	75	30	2	93.75
3	-noise -low -lpc -eucl	24	8	75	31	1	96.88
3	-noise -band -lpc -hamming	24	8	75	29	3	90.62
3	-noise -bandstop -fft -diff	24	8	75	32	0	100
3	-noise -endp -lpc -diff	24	8	75	32	0	100
3	-endp -lpc -eucl	24	8	75	30	2	93.75
3	-bandstop -aggr -cos	24	8	75	31	1	96.88
3	-low -lpc -diff	24	8	75	31	1	96.88
3	-silence -noise -low -aggr -eucl	24	8	75	32	0	100
3	-noise -norm -lpc -diff	24	8	75	31	1	96.88
3	-noise -low -lpc -diff	24	8	75	32	0	100
3	-endp -lpc -diff	24	8	75	30	2	93.75
3	-endp -lpc -cos	24	8	75	29	3	90.62
3	-silence -noise -band -lpc -cheb	24	8	75	32	0	100
3	-noise -endp -lpc -cos	24	8	75	31	1	96.88
3	-noise -endp -lpc -hamming	24	8	75	31	1	96.88
3	-noise -bandstop -aggr -cheb	24	8	75	32	0	100
3	-noise -bandstop -fft -cheb	24	8	75	32	0	100
3	-noise -endp -lpc -cheb	24	8	75	32	0	100
4	-noise -norm -lpc -cos	23	9	71.88	30	2	93.75
4	-silence -noise -band -lpc -eucl	23	9	71.88	32	0	100
4	-silence -noise -norm -aggr -cos	23	9	71.88	29	3	90.62
4	-silence -band -lpc -eucl	23	9	71.88	32	0	100
4	-silence -low -fft -cos	23	9	71.88	29	3	90.62
4	-noise -bandstop -fft -eucl	23	9	71.88	32	0	100
4	-silence -noise -norm -fft -cos	23	9	71.88	29	3	90.62
4	-raw -fft -eucl	23	9	71.88	32	0	100
4	-silence -noise -endp -lpc -hamming	23	9	71.88	31	1	96.88
4	-high -aggr -mink	23	9	71.88	32	0	100
4	-noise -low -aggr -diff	23	9	71.88	32	0	100
4	-low -fft -cos	23	9	71.88	29	3	90.62
4	-silence -noise -low -fft -cos	23	9	71.88	29	3	90.62
4	-silence -band -lpc -diff	23	9	71.88	31	1	96.88
4	-noise -bandstop -aggr -cos	23	9	71.88	29	3	90.62
4	-silence -noise -low -fft -diff	23	9	71.88	32	0	100
4	-bandstop -fft -eucl	23	9	71.88	32	0	100

Table 6. Top Most Accurate Configurations for Gender Identification, 1st and 2nd Guesses, Median Clustering (Mokhov (2008d))

Guess	Rank	Configuration	GOOD	BAD	Precision, %
1st	1	-wav -raw -lpc -cheb	147	54	73.13
1st	1	-wav -silence -noise -raw -lpc -cheb	147	54	73.13
1st	1	-wav -noise -raw -lpc -cheb	147	54	73.13
1st	1	-wav -norm -lpc -cheb	147	54	73.13
1st	1	-wav -silence -raw -lpc -cheb	147	54	73.13
1st	2	-wav -silence -norm -fft -cheb	129	72	64.18
1st	3	-wav -bandstop -fft -cheb	125	76	62.19
1st	3	-wav -silence -noise -norm -fft -cheb	125	76	62.19
1st	3	-wav -silence -low -fft -cheb	125	76	62.19
1st	4	-wav -silence -norm -lpc -cheb	124	77	61.69
1st	5	-wav -silence -noise -low -fft -cheb	122	79	60.70
1st	6	-wav -silence -noise -raw -lpc -cos	120	81	59.70
1st	6	-wav -noise -raw -lpc -cos	120	81	59.70
1st	6	-wav -raw -lpc -cos	120	81	59.70
1st	6	-wav -silence -raw -lpc -cos	120	81	59.70
1st	6	-wav -norm -lpc -cos	120	81	59.70
1st	7	-wav -noise -bandstop -fft -cheb	119	82	59.20
1st	7	-wav -silence -noise -bandstop -lpc -cos	119	82	59.20
1st	8	-wav -silence -noise -bandstop -lpc -cheb	118	83	58.71
1st	8	-wav -silence -norm -fft -cos	118	83	58.71
1st	8	-wav -silence -bandstop -fft -cheb	118	83	58.71
1st	9	-wav -bandstop -fft -cos	115	86	57.21
1st	10	-wav -silence -noise -bandstop -fft -cheb	112	89	55.72
1st	11	-wav -noise -raw -fft -cheb	111	90	55.22
1st	11	-wav -silence -noise -raw -fft -cheb	111	90	55.22
1st	11	-wav -silence -raw -fft -cheb	111	90	55.22
1st	11	-wav -raw -fft -cheb	111	90	55.22
1st	12	-wav -silence -noise -raw -fft -cos	110	91	54.73
1st	12	-wav -noise -raw -fft -cos	110	91	54.73
1st	12	-wav -raw -fft -cos	110	91	54.73
1st	12	-wav -silence -raw -fft -cos	110	91	54.73
1st	13	-wav -noise -bandstop -lpc -cos	109	92	54.23
1st	13	-wav -norm -fft -cos	109	92	54.23
1st	13	-wav -norm -fft -cheb	109	92	54.23
1st	14	-wav -silence -low -lpc -cheb	105	96	52.24
1st	14	-wav -silence -noise -norm -lpc -cheb	105	96	52.24
1st	15	-wav -silence -norm -lpc -cos	101	100	50.25
1st	16	-wav -silence -bandstop -fft -cos	99	102	49.25
1st	17	-wav -noise -norm -lpc -cos	96	105	47.76
1st	17	-wav -low -lpc -cos	96	105	47.76
1st	18	-wav -silence -noise -low -fft -cos	92	109	45.77
1st	19	-wav -noise -low -lpc -cos	91	110	45.27
1st	20	-wav -silence -noise -low -lpc -cheb	87	114	43.28
1st	20	-wav -silence -low -fft -cos	87	114	43.28
1st	20	-wav -silence -noise -norm -fft -cos	87	114	43.28
1st	21	-wav -noise -low -fft -cheb	86	115	42.79
1st	22	-wav -silence -low -lpc -cos	85	116	42.29
1st	22	-wav -silence -noise -norm -lpc -cos	85	116	42.29
1st	23	-wav -noise -low -fft -cos	84	117	41.79
1st	23	-wav -low -lpc -cheb	84	117	41.79
1st	23	-wav -noise -norm -lpc -cheb	84	117	41.79
1st	24	-wav -noise -low -lpc -cheb	82	119	40.80
1st	25	-wav -noise -norm -fft -cos	81	120	40.30
1st	25	-wav -low -fft -cos	81	120	40.30
1st	26	-wav -low -fft -cheb	80	121	39.80
1st	26	-wav -noise -norm -fft -cheb	80	121	39.80
1st	26	-wav -noise -bandstop -lpc -cheb	80	121	39.80
1st	27	-wav -silence -noise -bandstop -fft -cos	78	123	38.81
1st	28	-wav -silence -noise -low -lpc -cos	76	125	37.81
1st	29	-wav -noise -bandstop -fft -cos	75	126	37.31
1st	30	-wav -bandstop -lpc -cheb	74	127	36.82
1st	31	-wav -silence -bandstop -lpc -cheb	65	136	32.34
1st	32	-wav -bandstop -lpc -cos	63	138	31.34
1st	33	-wav -silence -bandstop -lpc -cos	54	147	26.87

Table 7. File types identification top results, bigrams (Mokhov & Debbabi (2008))

Certain results were quite encouraging for the first and second best statistics extracts in Table 7 and Table 8, as well as statistics per file type in Table 9. We also collected the worst statistics, where the use of a "raw" loader impacted negatively drastically the accuracy of the results as shown in Table 10 and Table 11; yet, some file types were robustly recognized, as shown in Table 12. This gives a clue to the researchers and investigators in which direction to follow to increase the precision and which ones not to use.

Guess	Rank	Configuration	GOOD	BAD	Precision, %
2nd	1	-wav -raw -lpc -cheb	166	35	82.59
2nd	1	-wav -silence -noise -raw -lpc -cheb	166	35	82.59
2nd	1	-wav -noise -raw -lpc -cheb	166	35	82.59
2nd	1	-wav -norm -lpc -cheb	166	35	82.59
2nd	1	-wav -silence -raw -lpc -cheb	166	35	82.59
2nd	2	-wav -silence -norm -fft -cheb	137	64	68.16
2nd	3	-wav -bandstop -fft -cheb	130	71	64.68
2nd	3	-wav -silence -noise -norm -fft -cheb	140	61	69.65
2nd	3	-wav -silence -low -fft -cheb	140	61	69.65
2nd	4	-wav -silence -norm -lpc -cheb	176	25	87.56
2nd	5	-wav -silence -noise -low -fft -cheb	142	59	70.65
2nd	6	-wav -silence -noise -raw -lpc -cos	142	59	70.65
2nd	6	-wav -noise -raw -lpc -cos	142	59	70.65
2nd	6	-wav -raw -lpc -cos	142	59	70.65
2nd	6	-wav -silence -raw -lpc -cos	142	59	70.65
2nd	6	-wav -norm -lpc -cos	142	59	70.65
2nd	7	-wav -noise -bandstop -fft -cheb	138	63	68.66
2nd	7	-wav -silence -noise -bandstop -lpc -cos	151	50	75.12
2nd	8	-wav -silence -noise -bandstop -lpc -cheb	156	45	77.61
2nd	8	-wav -silence -norm -fft -cos	147	54	73.13
2nd	8	-wav -silence -bandstop -fft -cheb	129	72	64.18
2nd	9	-wav -bandstop -fft -cos	127	74	63.18
2nd	10	-wav -silence -noise -bandstop -fft -cheb	135	66	67.16
2nd	11	-wav -noise -raw -fft -cheb	122	79	60.70
2nd	11	-wav -silence -noise -raw -fft -cheb	122	79	60.70
2nd	11	-wav -silence -raw -fft -cheb	122	79	60.70
2nd	11	-wav -raw -fft -cheb	122	79	60.70
2nd	12	-wav -silence -noise -raw -fft -cos	130	71	64.68
2nd	12	-wav -noise -raw -fft -cos	130	71	64.68
2nd	12	-wav -raw -fft -cos	130	71	64.68
2nd	12	-wav -silence -raw -fft -cos	130	71	64.68
2nd	13	-wav -noise -bandstop -lpc -cos	148	53	73.63
2nd	13	-wav -norm -fft -cos	130	71	64.68
2nd	13	-wav -norm -fft -cheb	121	80	60.20
2nd	14	-wav -silence -low -lpc -cheb	127	74	63.18
2nd	14	-wav -silence -noise -norm -lpc -cheb	127	74	63.18
2nd	15	-wav -silence -norm -lpc -cos	151	50	75.12
2nd	16	-wav -silence -bandstop -fft -cos	135	66	67.16
2nd	17	-wav -noise -norm -lpc -cos	118	83	58.71
2nd	17	-wav -low -lpc -cos	118	83	58.71
2nd	18	-wav -silence -noise -low -fft -cos	146	55	72.64
2nd	19	-wav -noise -low -lpc -cos	115	86	57.21
2nd	20	-wav -silence -noise -low -lpc -cheb	120	81	59.70
2nd	20	-wav -silence -low -fft -cos	143	58	71.14
2nd	20	-wav -silence -noise -norm -fft -cos	143	58	71.14
2nd	21	-wav -noise -low -fft -cheb	130	71	64.68
2nd	22	-wav -silence -low -lpc -cos	111	90	55.22
2nd	22	-wav -silence -noise -norm -lpc -cos	111	90	55.22
2nd	23	-wav -noise -low -fft -cos	128	73	63.68
2nd	23	-wav -low -lpc -cheb	130	71	64.68
2nd	23	-wav -noise -norm -lpc -cheb	130	71	64.68
2nd	24	-wav -noise -low -lpc -cheb	129	72	64.18
2nd	25	-wav -noise -norm -fft -cos	129	72	64.18
2nd	25	-wav -low -fft -cos	129	72	64.18
2nd	26	-wav -low -fft -cheb	115	86	57.21
2nd	26	-wav -noise -norm -fft -cheb	115	86	57.21
2nd	26	-wav -noise -bandstop -lpc -cheb	127	74	63.18
2nd	27	-wav -silence -noise -bandstop -fft -cos	125	76	62.19
2nd	28	-wav -silence -noise -low -lpc -cos	118	83	58.71
2nd	29	-wav -noise -bandstop -fft -cos	123	78	61.19
2nd	30	-wav -bandstop -lpc -cheb	111	90	55.22
2nd	31	-wav -silence -bandstop -lpc -cheb	133	68	66.17
2nd	32	-wav -bandstop -lpc -cos	123	78	61.19
2nd	33	-wav -silence -bandstop -lpc -cos	126	75	62.69

Table 8. File types identification top results, 2nd best, bigrams (Mokhov & Debbabi (2008))

In addition to the previously described options, here we also have: `-wav` that corresponds to a custom loader that translates any files into a WAV-like format. The detail that is not present in the resulting tables are the internal configuration of the loader's *n*-grams loading or raw state.

3. The results in Table 13 represent the classification of the French publications using the same spectral techniques to determine whether a particular article in the French press was published in France or Quebec. The complete description of the related experiments and results can be found in Mokhov (2010a;b).
   In addition to the previously mentioned options, we have: `-title-only` to indicate to work with article titles only instead of main body texts; `-ref` tells the system to validate against reference data supplied by the organizers rather than the training data.

Guess	Rank	File type	GOOD	BAD	Precision, %
1st	1	Mach-O filetype=10 i386	64	0	100.00
1st	2	HTML document text	64	0	100.00
1st	3	TIFF image data; big-endian	64	0	100.00
1st	4	data	64	0	100.00
1st	5	ASCII c program text; with very long lines	64	0	100.00
1st	6	Rich Text Format data; version 1; Apple Macintosh	128	0	100.00
1st	7	ASCII English text	64	0	100.00
1st	8	a /sw/bin/ocamlrun script text executable	516	60	89.58
1st	9	perl script text executable	832	192	81.25
1st	10	NeXT/Apple typedstream data; big endian; version 4; system 1000	255	65	79.69
1st	11	Macintosh Application (data)	48	16	75.00
1st	12	XML 1.0 document text	320	128	71.43
1st	13	ASCII text	242	142	63.02
1st	14	Mach-O executable i386	3651	3325	52.34
1st	15	Bourne shell script text executable	262	2298	10.23
2nd	1	Mach-O filetype=10 i386	64	0	100.00
2nd	2	HTML document text	64	0	100.00
2nd	3	TIFF image data; big-endian	64	0	100.00
2nd	4	data	64	0	100.00
2nd	5	ASCII c program text; with very long lines	64	0	100.00
2nd	6	Rich Text Format data; version 1; Apple Macintosh	128	0	100.00
2nd	7	ASCII English text	64	0	100.00
2nd	8	a /sw/bin/ocamlrun script text executable	529	47	91.84
2nd	9	perl script text executable	960	64	93.75
2nd	10	NeXT/Apple typedstream data; big endian; version 4; system 1000	281	39	87.81
2nd	11	Macintosh Application (data)	64	0	100.00
2nd	12	XML 1.0 document text	366	82	81.70
2nd	13	ASCII text	250	134	65.10
2nd	14	Mach-O executable i386	5091	1885	72.98
2nd	15	Bourne shell script text executable	528	2032	20.62

Table 9. File types identification top results, bigrams, per file type (Mokhov & Debbabi (2008))

8. Conclusion

We presented an overview of MARF, a modular and extensible pattern recognition framework for a reasonably diverse spectrum of the learning and recognition tasks. We outlined the pipeline and the data structures used in this open-source project in a practical manner. We provided some typical results one can obtain by running MARF's implementations for various learning and classification problems.

8.1 Advantages and disadvantages of the approach

The framework approach is both an advantage and a disadvantage. The advantage is obvious – a consistent and uniform environment and implementing platform for comparative studies with a plug-in architecture. However, as the number of algorithms grows it is more difficult to adjust the framework's API itself without breaking all the modules that depend on it.

The coverage of algorithms is as good as the number of them implemented in / contributed to the project. In the results mentioned in Section 7 we could have attained better precision in some cases if better algorithm implementations were available (or any bugs in exiting ones fixed).

Guess	Rank	Configuration	GOOD	BAD	Precision, %
1st	1	-wav -noise -raw -fft -cheb	9	192	4.48
1st	1	-wav -raw -lpc -cheb	9	192	4.48
1st	1	-wav -bandstop -fft -cheb	9	192	4.48
1st	1	-wav -noise -low -fft -cos	9	192	4.48
1st	1	-wav -noise -norm -fft -cos	9	192	4.48
1st	1	-wav -noise -low -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -raw -lpc -cheb	9	192	4.48
1st	1	-wav -low -fft -cos	9	192	4.48
1st	1	-wav -silence -noise -raw -fft -cos	9	192	4.48
1st	1	-wav -noise -low -lpc -cos	9	192	4.48
1st	1	-wav -silence -noise -low -lpc -cheb	9	192	4.48
1st	1	-wav -noise -bandstop -lpc -cos	9	192	4.48
1st	1	-wav -noise -norm -lpc -cos	9	192	4.48
1st	1	-wav -silence -low -fft -cos	9	192	4.48
1st	1	-wav -silence -noise -raw -fft -cheb	9	192	4.48
1st	1	-wav -silence -low -lpc -cheb	9	192	4.48
1st	1	-wav -silence -noise -norm -fft -cheb	9	192	4.48
1st	1	-wav -silence -raw -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -bandstop -lpc -cheb	9	192	4.48
1st	1	-wav -noise -raw -fft -cos	9	192	4.48
1st	1	-wav -low -lpc -cos	9	192	4.48
1st	1	-wav -silence -noise -bandstop -fft -cos	9	192	4.48
1st	1	-wav -silence -norm -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -raw -lpc -cos	9	192	4.48
1st	1	-wav -silence -norm -fft -cos	9	192	4.48
1st	1	-wav -raw -fft -cos	9	192	4.48
1st	1	-wav -silence -low -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -low -fft -cos	9	192	4.48
1st	1	-wav -silence -bandstop -lpc -cos	9	192	4.48
1st	1	-wav -bandstop -fft -cos	9	192	4.48
1st	1	-wav -noise -raw -lpc -cos	9	192	4.48
1st	1	-wav -noise -bandstop -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -bandstop -lpc -cos	9	192	4.48
1st	1	-wav -silence -raw -fft -cos	9	192	4.48
1st	1	-wav -raw -lpc -cos	9	192	4.48
1st	1	-wav -silence -norm -lpc -cos	9	192	4.48
1st	1	-wav -silence -noise -low -lpc -cos	9	192	4.48
1st	1	-wav -noise -raw -lpc -cheb	9	192	4.48
1st	1	-wav -low -lpc -cheb	9	192	4.48
1st	1	-wav -raw -fft -cheb	9	192	4.48
1st	1	-wav -silence -bandstop -lpc -cheb	9	192	4.48
1st	1	-wav -norm -lpc -cheb	9	192	4.48
1st	1	-wav -silence -raw -lpc -cos	9	192	4.48
1st	1	-wav -noise -low -lpc -cheb	9	192	4.48
1st	1	-wav -noise -norm -lpc -cheb	9	192	4.48
1st	1	-wav -norm -fft -cos	9	192	4.48
1st	1	-wav -low -fft -cheb	9	192	4.48
1st	1	-wav -silence -bandstop -fft -cheb	9	192	4.48
1st	1	-wav -norm -fft -cheb	9	192	4.48
1st	1	-wav -noise -bandstop -fft -cos	9	192	4.48
1st	1	-wav -noise -norm -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -norm -fft -cos	9	192	4.48
1st	1	-wav -silence -noise -low -fft -cheb	9	192	4.48
1st	1	-wav -silence -noise -norm -lpc -cheb	9	192	4.48
1st	1	-wav -norm -lpc -cos	9	192	4.48
1st	1	-wav -silence -raw -lpc -cheb	9	192	4.48
1st	1	-wav -silence -noise -bandstop -fft -cheb	9	192	4.48
1st	1	-wav -silence -low -lpc -cos	9	192	4.48
1st	1	-wav -silence -norm -lpc -cheb	9	192	4.48
1st	1	-wav -silence -bandstop -fft -cos	9	192	4.48
1st	1	-wav -silence -noise -norm -lpc -cos	9	192	4.48
1st	1	-wav -noise -bandstop -lpc -cheb	9	192	4.48
1st	1	-wav -bandstop -lpc -cos	9	192	4.48
1st	1	-wav -bandstop -lpc -cheb	9	192	4.48

Table 10. File types identification worst results, raw loader (Mokhov & Debbabi (2008))

8.2 Future work

The general goals of the future and ongoing research include:

- There are a lot more algorithms to implement and test for the existing tasks.
- Apply to more case studies.
- Enhance statistics reporting and details thereof (memory usage, run-time, recall, f-measure, etc.).

- Scalability studies with the General Intensional Programming System (GIPSY) project (see Mokhov & Paquet (2010); Paquet (2009); Paquet & Wu (2005); The GIPSY Research and Development Group (2002–2010); Vassev & Paquet (2008)).

Guess	Rank	Configuration	GOOD	BAD	Precision, %
2nd	1	-wav -noise -raw -fft -cheb	10	191	4.98
2nd	1	-wav -raw -lpc -cheb	10	191	4.98
2nd	1	-wav -bandstop -fft -cheb	10	191	4.98
2nd	1	-wav -noise -low -fft -cos	10	191	4.98
2nd	1	-wav -noise -norm -fft -cos	10	191	4.98
2nd	1	-wav -noise -low -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -raw -lpc -cheb	10	191	4.98
2nd	1	-wav -low -fft -cos	10	191	4.98
2nd	1	-wav -silence -noise -raw -fft -cos	10	191	4.98
2nd	1	-wav -noise -low -lpc -cos	10	191	4.98
2nd	1	-wav -silence -noise -low -lpc -cheb	10	191	4.98
2nd	1	-wav -noise -bandstop -lpc -cos	10	191	4.98
2nd	1	-wav -noise -norm -lpc -cos	10	191	4.98
2nd	1	-wav -silence -low -fft -cos	10	191	4.98
2nd	1	-wav -silence -noise -raw -fft -cheb	10	191	4.98
2nd	1	-wav -silence -low -lpc -cheb	10	191	4.98
2nd	1	-wav -silence -noise -norm -fft -cheb	10	191	4.98
2nd	1	-wav -silence -raw -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -bandstop -lpc -cheb	10	191	4.98
2nd	1	-wav -noise -raw -fft -cos	10	191	4.98
2nd	1	-wav -low -lpc -cos	10	191	4.98
2nd	1	-wav -silence -noise -bandstop -fft -cos	10	191	4.98
2nd	1	-wav -silence -norm -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -raw -lpc -cos	10	191	4.98
2nd	1	-wav -silence -norm -fft -cos	10	191	4.98
2nd	1	-wav -raw -fft -cos	10	191	4.98
2nd	1	-wav -silence -low -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -low -fft -cos	10	191	4.98
2nd	1	-wav -silence -bandstop -lpc -cos	10	191	4.98
2nd	1	-wav -bandstop -fft -cos	10	191	4.98
2nd	1	-wav -noise -raw -lpc -cos	10	191	4.98
2nd	1	-wav -noise -bandstop -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -bandstop -lpc -cos	10	191	4.98
2nd	1	-wav -silence -raw -fft -cos	10	191	4.98
2nd	1	-wav -raw -lpc -cos	10	191	4.98
2nd	1	-wav -silence -norm -lpc -cos	10	191	4.98
2nd	1	-wav -silence -noise -low -lpc -cos	10	191	4.98
2nd	1	-wav -noise -raw -lpc -cheb	10	191	4.98
2nd	1	-wav -low -lpc -cheb	10	191	4.98
2nd	1	-wav -raw -fft -cheb	10	191	4.98
2nd	1	-wav -silence -bandstop -lpc -cheb	10	191	4.98
2nd	1	-wav -norm -lpc -cheb	10	191	4.98
2nd	1	-wav -silence -raw -lpc -cos	10	191	4.98
2nd	1	-wav -noise -low -lpc -cheb	10	191	4.98
2nd	1	-wav -noise -norm -lpc -cheb	10	191	4.98
2nd	1	-wav -norm -fft -cos	10	191	4.98
2nd	1	-wav -low -fft -cheb	10	191	4.98
2nd	1	-wav -silence -bandstop -fft -cheb	10	191	4.98
2nd	1	-wav -norm -fft -cheb	10	191	4.98
2nd	1	-wav -noise -bandstop -fft -cos	10	191	4.98
2nd	1	-wav -noise -norm -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -norm -fft -cos	10	191	4.98
2nd	1	-wav -silence -noise -low -fft -cheb	10	191	4.98
2nd	1	-wav -silence -noise -norm -lpc -cheb	10	191	4.98
2nd	1	-wav -norm -lpc -cos	10	191	4.98
2nd	1	-wav -silence -raw -lpc -cheb	10	191	4.98
2nd	1	-wav -silence -noise -bandstop -fft -cheb	10	191	4.98
2nd	1	-wav -silence -low -lpc -cos	10	191	4.98
2nd	1	-wav -silence -norm -lpc -cheb	10	191	4.98
2nd	1	-wav -silence -bandstop -fft -cos	10	191	4.98
2nd	1	-wav -silence -noise -norm -lpc -cos	10	191	4.98
2nd	1	-wav -noise -bandstop -lpc -cheb	10	191	4.98
2nd	1	-wav -bandstop -lpc -cos	10	191	4.98
2nd	1	-wav -bandstop -lpc -cheb	10	191	4.98

Table 11. File types identification worst results, 2nd guess, raw loader (Mokhov & Debbabi (2008))

Guess	Rank	File type	GOOD	BAD	Precision, %
1st	1	a /sw/bin/ocamlrun script text executable	576	0	100.00
1st	2	Bourne shell script text executable	0	2560	0.00
1st	3	Mach-O filetype=10 i386	0	64	0.00
1st	4	HTML document text	0	64	0.00
1st	5	NeXT/Apple typedstream data; big endian; version 4; system 1000	0	320	0.00
1st	6	Mach-O executable i386	0	6976	0.00
1st	7	ASCII text	0	384	0.00
1st	8	TIFF image data; big-endian	0	64	0.00
1st	9	Macintosh Application (data)	0	64	0.00
1st	10	data	0	64	0.00
1st	11	ASCII c program text; with very long lines	0	64	0.00
1st	12	perl script text executable	0	1024	0.00
1st	13	Rich Text Format data; version 1; Apple Macintosh	0	128	0.00
1st	14	XML 1.0 document text	0	448	0.00
1st	15	ASCII English text	0	64	0.00
2nd	1	a /sw/bin/ocamlrun script text executable	576	0	100.00
2nd	2	Bourne shell script text executable	0	2560	0.00
2nd	3	Mach-O filetype=10 i386	0	64	0.00
2nd	4	HTML document text	0	64	0.00
2nd	5	NeXT/Apple typedstream data; big endian; version 4; system 1000	0	320	0.00
2nd	6	Mach-O executable i386	0	6976	0.00
2nd	7	ASCII text	0	384	0.00
2nd	8	TIFF image data; big-endian	0	64	0.00
2nd	9	Macintosh Application (data)	64	0	100.00
2nd	10	data	0	64	0.00
2nd	11	ASCII c program text; with very long lines	0	64	0.00
2nd	12	perl script text executable	0	1024	0.00
2nd	13	Rich Text Format data; version 1; Apple Macintosh	0	128	0.00
2nd	14	XML 1.0 document text	0	448	0.00
2nd	15	ASCII English text	0	64	0.00

Table 12. File types identification worst results, per file, raw loader (Mokhov & Debbabi (2008))

9. Acknowledgments

This work was funded in part by the Faculty of Engineering and Computer Science (ENCS), Concordia University, Montreal, Canada. We would like to acknowledge the original co-creators of MARF: Stephen Sinclair, Ian Clément, Dimitrios Nicolacopoulos as well as subsequent contributors of the MARF R&D Group, including Lee Wei Huynh, Jian "James" Li, Farid Rassai, and all other contributors. The author would like to also mention the people who inspired some portions of this or the related work including Drs. Leila Kosseim, Sabine Bergler, Ching Y. Suen, Lingyu Wang, Joey Paquet, Mourad Debbabi, Amr M. Youssef, Chadi M. Assi, Emil Vassev, Javad Sadri; and Michelle Khalifé.

10. References

Abdi, H. (2007). Distance, *in* N. J. Salkind (ed.), *Encyclopedia of Measurement and Statistics*, Thousand Oaks (CA): Sage.

Bernsee, S. M. (1999–2005). The DFT "à pied": Mastering the Fourier transform in one day, [online]. http://www.dspdimension.com/data/html/dftapied.html.

Cavalin, P. R., Sabourin, R. & Suen, C. Y. (2010). Dynamic selection of ensembles of classifiers using contextual information, *Multiple Classifier Systems*, LNCS 5997, pp. 145–154.

Clement, I., Mokhov, S. A., Nicolacopoulos, D., Fan, S. & the MARF Research & Development Group (2002–2010). TestLPC – Testing LPC Algorithm Implementation within MARF, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Rank #	Guess	Configuration	GOOD	BAD	Precision,%
1	1st	-title-only -ref -silence -noise -norm -aggr -eucl	1714	768	69.06
1	1st	-title-only -ref -silence -noise -norm -fft -eucl	1714	768	69.06
1	1st	-title-only -ref -low -aggr -eucl	1714	768	69.06
1	1st	-title-only -ref -noise -norm -aggr -eucl	1714	768	69.06
1	1st	-title-only -ref -silence -low -aggr -eucl	1714	768	69.06
1	1st	-title-only -ref -noise -norm -fft -eucl	1714	768	69.06
1	1st	-title-only -ref -silence -low -fft -eucl	1714	768	69.06
1	1st	-title-only -ref -low -fft -eucl	1714	768	69.06
2	1st	-title-only -ref -noise -endp -fft -eucl	1701	781	68.53
2	1st	-title-only -ref -noise -endp -aggr -eucl	1701	781	68.53
2	1st	-title-only -ref -silence -noise -endp -fft -eucl	1701	781	68.53
2	1st	-title-only -ref -silence -noise -endp -aggr -eucl	1701	781	68.53
3	1st	-title-only -ref -silence -noise -bandstop -aggr -eucl	1694	788	68.25
3	1st	-title-only -ref -silence -noise -bandstop -fft -eucl	1694	788	68.25
3	1st	-title-only -ref -noise -bandstop -aggr -eucl	1694	788	68.25
3	1st	-title-only -ref -noise -bandstop -fft -eucl	1694	788	68.25
4	1st	-title-only -ref -bandstop -aggr -cos	1691	791	68.13
4	1st	-title-only -ref -bandstop -fft -cos	1691	791	68.13
5	1st	-title-only -ref -silence -bandstop -fft -cos	1690	792	68.09
5	1st	-title-only -ref -silence -bandstop -aggr -cos	1690	792	68.09
6	1st	-title-only -ref -bandstop -fft -eucl	1688	794	68.01
6	1st	-title-only -ref -bandstop -aggr -eucl	1688	794	68.01
7	1st	-title-only -ref -silence -bandstop -fft -eucl	1686	796	67.93
7	1st	-title-only -ref -silence -bandstop -aggr -eucl	1686	796	67.93
8	1st	-title-only -ref -norm -fft -eucl	1678	804	67.61
8	1st	-title-only -ref -norm -aggr -cos	1678	804	67.61
8	1st	-title-only -ref -silence -norm -fft -cos	1678	804	67.61
8	1st	-title-only -ref -norm -aggr -eucl	1678	804	67.61
8	1st	-title-only -ref -norm -fft -cos	1678	804	67.61
8	1st	-title-only -ref -silence -norm -aggr -eucl	1678	804	67.61
8	1st	-title-only -ref -silence -norm -fft -eucl	1678	804	67.61
8	1st	-title-only -ref -silence -norm -aggr -cos	1678	804	67.61
9	1st	-title-only -ref -silence -raw -fft -eucl	1676	806	67.53
9	1st	-title-only -ref -silence -raw -aggr -eucl	1676	806	67.53
9	1st	-title-only -ref -raw -fft -eucl	1676	806	67.53
9	1st	-title-only -ref -noise -raw -fft -eucl	1676	806	67.53
9	1st	-title-only -ref -noise -raw -aggr -eucl	1676	806	67.53
9	1st	-title-only -ref -silence -noise -raw -aggr -eucl	1676	806	67.53
9	1st	-title-only -ref -silence -noise -raw -fft -eucl	1676	806	67.53
9	1st	-title-only -ref -raw -aggr -eucl	1676	806	67.53
10	1st	-title-only -ref -silence -noise -low -aggr -eucl	1670	812	67.28
10	1st	-title-only -ref -silence -noise -low -fft -eucl	1670	812	67.28
11	1st	-title-only -ref -noise -low -fft -eucl	1669	813	67.24
11	1st	-title-only -ref -noise -low -aggr -eucl	1669	813	67.24
12	1st	-title-only -ref -endp -fft -cos	1651	831	66.52
13	1st	-title-only -ref -silence -low -aggr -cheb	1631	851	65.71
13	1st	-title-only -ref -silence -low -fft -cheb	1631	851	65.71
13	1st	-title-only -ref -silence -noise -norm -aggr -cheb	1631	851	65.71
13	1st	-title-only -ref -silence -noise -norm -fft -cheb	1631	851	65.71

Table 13. Geographic location identification using article titles only on reference data (Mokhov (2010b))

Clement, I., Mokhov, S. A. & the MARF Research & Development Group (2002–2010). TestNN – Testing Artificial Neural Network in MARF, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Darwin, I. F., Gilmore, J., Collyer, G., McMahon, R., Harris, G., Zoulas, C., Lowth, C., Fischer, E. & Various Contributors (1973–2007). `file` – determine file type, BSD General Commands Manual, file(1), BSD. man file(1).

Darwin, I. F., Gilmore, J., Collyer, G., McMahon, R., Harris, G., Zoulas, C., Lowth, C., Fischer, E. & Various Contributors (1973–2008). `file` – determine file type, [online]. ftp: //ftp.astron.com/pub/file/, last viewed April 2008.

Flanagan, D. (1997). *Java in a Nutshell*, second edn, O'Reily & Associates, Inc. ISBN 1-56592-262-X.

Forest, D., Grouin, C., Sylva, L. D. & DEFT (2010). Campagne DÉfi Fouille de Textes (DEFT) 2010, [online], http://www.groupes.polymtl.ca/taln2010/deft.php.

Freeman, E., Freeman, E., Sierra, K. & Bates, B. (2004). *Head First Design Patterns*, first edn, O'Reilly. http://www.oreilly.com/catalog/hfdesignpat/toc.pdf, http://www.oreilly.com/catalog/hfdesignpat/chapter/index.html.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley. ISBN: 0201633612.

Garcia, E. (2006). Cosine similarity and term weight tutorial, [online]. http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html.

Green, D. (2001–2005). Java reflection API, Sun Microsystems, Inc. http://java.sun.com/docs/books/tutorial/reflect/index.html.

Hamming, R. W. (1950). Error detecting and error correcting codes, *Bell System Technical Journal* 26(2): 147–160. See also http://en.wikipedia.org/wiki/Hamming_ distance.

Haridas, S. (2006). *Generation of 2-D digital filters with variable magnitude characteristics starting from a particular type of 2-variable continued fraction expansion*, Master's thesis, Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada.

Haykin, S. (1988). *Digital Communications*, John Wiley and Sons, New York, NY, USA.

Ifeachor, E. C. & Jervis, B. W. (2002). *Speech Communications*, Prentice Hall, New Jersey, USA.

Jini Community (2007). Jini network technology, [online]. http://java.sun.com/developer/products/jini/index.jsp.

Jurafsky, D. S. & Martin, J. H. (2000). *Speech and Language Processing*, Prentice-Hall, Inc., Pearson Higher Education, Upper Saddle River, New Jersey 07458. ISBN 0-13-095069-6.

Khalifé, M. (2004). *Examining orthogonal concepts-based micro-classifiers and their correlations with noun-phrase coreference chains*, Master's thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.

Larman, C. (2006). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, third edn, Pearson Education. ISBN: 0131489062.

Mahalanobis, P. C. (1936). On the generalised distance in statistics, *Proceedings of the National Institute of Science of India 12*, pp. 49–55. Online at http://en.wikipedia.org/wiki/Mahalanobis_distance.

Merx, G. G. & Norman, R. J. (2007). *Unified Software Engineering with Java*, Pearson Prentice Hall. ISBN: 978-0-13-047376-6.

Mokhov, S. A. (2006). On design and implementation of distributed modular audio recognition framework: Requirements and specification design document, [online]. Project report, http://arxiv.org/abs/0905.2459, last viewed April 2010.

Mokhov, S. A. (2007a). Introducing MARF: a modular audio recognition framework and its applications for scientific and software engineering research, *Advances in Computer and Information Sciences and Engineering*, Springer Netherlands, University of Bridgeport, U.S.A., pp. 473–478. Proceedings of CISSE/SCSS'07.

Mokhov, S. A. (2007b). MARF for PureData for MARF, *Pd Convention '07*, artengine.ca, Montreal, Quebec, Canada. http://artengine.ca/~catalogue-pd/32-Mokhov.pdf.

Mokhov, S. A. (2008–2010c). WriterIdentApp – Writer Identification Application, Unpublished.

Mokhov, S. A. (2008a). Choosing best algorithm combinations for speech processing tasks in machine learning using MARF, *in* S. Bergler (ed.), *Proceedings of the 21st Canadian AI'08*, Springer-Verlag, Berlin Heidelberg, Windsor, Ontario, Canada, pp. 216–221. LNAI 5032.

Mokhov, S. A. (2008b). Encoding forensic multimedia evidence from MARF applications as Forensic Lucid expressions, *in* T. Sobh, K. Elleithy & A. Mahmood (eds), *Novel Algorithms and Techniques in Telecommunications and Networking, proceedings of CISSE'08*, Springer, University of Bridgeport, CT, USA, pp. 413–416. Printed in January 2010.

Mokhov, S. A. (2008c). Experimental results and statistics in the implementation of the modular audio recognition framework's API for text-independent speaker identification, *in* C. D. Zinn, H.-W. Chu, M. Savoie, J. Ferrer & A. Munitic (eds), *Proceedings of the 6th International Conference on Computing, Communications and Control Technologies (CCCT'08)*, Vol. II, IIIS, Orlando, Florida, USA, pp. 267–272.

Mokhov, S. A. (2008d). Study of best algorithm combinations for speech processing tasks in machine learning using median vs. mean clusters in MARF, *in* B. C. Desai (ed.), *Proceedings of C3S2E'08*, ACM, Montreal, Quebec, Canada, pp. 29–43. ISBN 978-1-60558-101-9.

Mokhov, S. A. (2008e). Towards security hardening of scientific distributed demand-driven and pipelined computing systems, *Proceedings of the 7th International Symposium on Parallel and Distributed Computing (ISPDC'08)*, IEEE Computer Society, pp. 375–382.

Mokhov, S. A. (2008f). Towards syntax and semantics of hierarchical contexts in multimedia processing applications using MARFL, *Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC)*, IEEE Computer Society, Turku, Finland, pp. 1288–1294.

Mokhov, S. A. (2010a). Complete complimentary results report of the MARF's NLP approach to the DEFT 2010 competition, [online]. http://arxiv.org/abs/1006.3787.

Mokhov, S. A. (2010b). L'approche MARF à DEFT 2010: A MARF approach to DEFT 2010, *Proceedings of TALN'10*. To appear in DEFT 2010 System competition at TALN 2010.

Mokhov, S. A. & Debbabi, M. (2008). File type analysis using signal processing techniques and machine learning vs. `file` unix utility for forensic analysis, *in* O. Goebel, S. Frings, D. Guenther, J. Nedon & D. Schadt (eds), *Proceedings of the IT Incident Management and IT Forensics (IMF'08)*, GI, Mannheim, Germany, pp. 73–85. LNI140.

Mokhov, S. A., Fan, S. & the MARF Research & Development Group (2002–2010b). TestFilters – Testing Filters Framework of MARF, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Mokhov, S. A., Fan, S. & the MARF Research & Development Group (2005–2010a). Math-TestApp – Testing Normal and Complex Linear Algebra in MARF, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Mokhov, S. A., Huynh, L.W. & Li, J. (2007). Managing distributed MARF with SNMP, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada. Project Report. Hosted at http://marf.sf.net, last viewed April 2008.

Mokhov, S. A., Huynh, L.W. & Li, J. (2008). Managing distributed MARF's nodes with SNMP, *Proceedings of PDPTA'2008*, Vol. II, CSREA Press, Las Vegas, USA, pp. 948–954.

Mokhov, S. A., Huynh, L.W., Li, J. & Rassai, F. (2007). A Java Data Security Framework (JDSF) for MARF and HSQLDB, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada. Project report. Hosted at http://marf.sf.net, last viewed April 2008.

Mokhov, S. A. & Jayakumar, R. (2008). Distributed modular audio recognition framework (DMARF) and its applications over web services, *in* T. Sobh, K. Elleithy & A. Mahmood (eds), *Proceedings of TeNe'08*, Springer, University of Bridgeport, CT, USA, pp. 417–422. Printed in January 2010.

Mokhov, S. A., Miladinova, M., Ormandjieva, O., Fang, F. & Amirghahari, A. (2008–2010). Application of reverse requirements engineering to open-source, student, and legacy software systems. Unpublished.

Mokhov, S. A. & Paquet, J. (2010). Using the General Intensional Programming System (GIPSY) for evaluation of higher-order intensional logic (HOIL) expressions, *Proceedings of SERA 2010*, IEEE Computer Society, pp. 101–109. Online at http: //arxiv.org/abs/0906.3911.

Mokhov, S. A., Sinclair, S., Clement, I., Nicolacopoulos, D. & the MARF Research & Development Group (2002–2010). SpeakerIdentApp – Text-Independent Speaker Identification Application, Published electronically within the MARF project, http: //marf.sf.net. Last viewed February 2010.

Mokhov, S. A., Song, M. & Suen, C. Y. (2009). Writer identification using inexpensive signal processing techniques, *in* T. Sobh&K. Elleithy (eds), *Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09*, Springer, pp. 437–441. ISBN: 978- 90-481-9111-6, online at: http://arxiv.org/abs/0912.5502.

Mokhov, S. A. & the MARF Research & Development Group (2003–2010a). LangIdentApp – Language Identification Application, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Mokhov, S. A. & the MARF Research & Development Group (2003–2010b). Probabilistic-ParsingApp – Probabilistic NLP Parsing Application, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Mokhov, S. A. & Vassev, E. (2009a). Autonomic specification of self-protection for Distributed MARF with ASSL, *Proceedings of C3S2E'09*, ACM, New York, NY, USA, pp. 175–183.

Mokhov, S. A. & Vassev, E. (2009b). Leveraging MARF for the simulation of the securing maritime borders intelligent systems challenge, *Proceedings of the Huntsville Simulation Conference (HSC'09)*, SCS. To appear.

Mokhov, S. A. & Vassev, E. (2009c). Self-forensics through case studies of small to medium software systems, *Proceedings of IMF'09*, IEEE Computer Society, pp. 128–141.

Mokhov, S. A., Clement, I., Sinclair, S. & Nicolacopoulos, D. (2002–2003). Modular Audio Recognition Framework, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada. Project report, http://marf.sf.net, last viewed April 2010.

O'Shaughnessy, D. (2000). *Speech Communications*, IEEE, New Jersey, USA.

Paquet, J. (2009). Distributed eductive execution of hybrid intensional programs, *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC'09)*, IEEE Computer Society, Seattle, Washington, USA, pp. 218–224.

Paquet, J. & Wu, A. H. (2005). GIPSY – a platform for the investigation on intensional programming languages, *Proceedings of the 2005 International Conference on Programming Languages and Compilers (PLC 2005)*, CSREA Press, pp. 8–14.

Press, W. H. (1993). *Numerical Recipes in C*, second edn, Cambridge University Press, Cambridge, UK.

Puckette, M. & PD Community (2007–2010). Pure Data, [online]. http://puredata.org.

Russell, S. J. & Norvig, P. (eds) (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey, USA. ISBN 0-13-103805-2.

Sinclair, S., Mokhov, S. A., Nicolacopoulos, D., Fan, S. & the MARF Research & Development Group (2002–2010). TestFFT – Testing FFT Algorithm Implementation within MARF, Published electronically within the MARF project, http://marf.sf.net. Last viewed February 2010.

Sun Microsystems, Inc. (1994–2009). The Java website, Sun Microsystems, Inc. http://java.sun.com, viewed in April 2009.

Sun Microsystems, Inc. (2004). Java IDL, Sun Microsystems, Inc. http://java.sun.com/j2se/1.5.0/docs/guide/idl/index.html.

Sun Microsystems, Inc. (2006). The java web services tutorial (for Java Web Services Developer's Pack, v2.0), Sun Microsystems, Inc. http://java.sun.com/webservices/docs/2.0/tutorial/doc/index.html.

The GIPSY Research and Development Group (2002–2010). The General Intensional Programming System (GIPSY) project, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada. http://newton.cs.concordia.ca/~gipsy/, last viewed February 2010.

The MARF Research and Development Group (2002–2010). The Modular Audio Recognition Framework and its Applications, [online]. http://marf.sf.net and http://arxiv.org/abs/0905.1235, last viewed April 2010.

The Sphinx Group at Carnegie Mellon (2007–2010). The CMU Sphinx group open source speech recognition engines, [online]. http://cmusphinx.sourceforge.net.

Vaillant, P., Nock, R. & Henry, C. (2006). Analyse spectrale des textes: détection automatique des frontières de langue et de discours, *Verbum ex machina: Actes de la 13eme conference annuelle sur le Traitement Automatique des Langues Naturelles (TALN 2006)*, pp. 619–629. Online at http://arxiv.org/abs/0810.1212.

Vassev, E. & Mokhov, S. A. (2009). Self-optimization property in autonomic specification of Distributed MARF with ASSL, *in* B. Shishkov, J. Cordeiro & A. Ranchordas (eds), *Proceedings of ICSOFT'09*, Vol. 1, INSTICC Press, Sofia, Bulgaria, pp. 331–335.

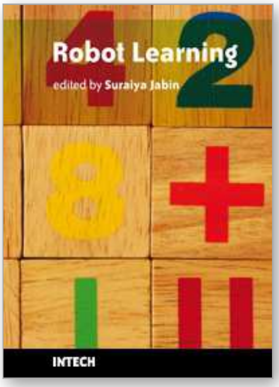
Vassev, E. & Mokhov, S. A. (2010). Towards autonomic specification of Distributed MARF with ASSL: Self-healing, *Proceedings of SERA 2010*, Vol. 296 of *SCI*, Springer, pp. 1–15.

Vassev, E. & Paquet, J. (2008). Towards autonomic GIPSY, *Proceedings of the Fifth IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASE 2008)*, IEEE Computer Society, pp. 25–34.

Wollrath, A. & Waldo, J. (1995–2005). Java RMI tutorial, Sun Microsystems, Inc. http://java.sun.com/docs/books/tutorial/rmi/index.html.

Zipf, G. K. (1935). *The Psychobiology of Language*, Houghton-Mifflin, New York, NY. See also http://en.wikipedia.org/wiki/Zipf%27s_law.

Zwicker, E. & Fastl, H. (1990). *Psychoacoustics facts and models*, Springer-Verlag, Berlin.



Robot Learning
Edited by Suraiya Jabin

ISBN 978-953-307-104-6
Hard cover, 150 pages
Publisher Sciyo
Published online 12, August, 2010
Published in print edition August, 2010

Robot Learning is intended for one term advanced Machine Learning courses taken by students from different computer science research disciplines. This text has all the features of a renowned best selling text. It gives a focused introduction to the primary themes in a Robot learning course and demonstrates the relevance and practicality of various Machine Learning algorithms to a wide variety of real-world applications from evolutionary techniques to reinforcement learning, classification, control, uncertainty and many other important fields. Salient features: - Comprehensive coverage of Evolutionary Techniques, Reinforcement Learning and Uncertainty. - Precise mathematical language used without excessive formalism and abstraction. - Included applications demonstrate the utility of the subject in terms of real-world problems. - A separate chapter on Anticipatory-mechanisms-of-human-sensory-motor-coordination and biped locomotion. - Collection of most recent research on Robot Learning.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Serguei Mokhov (2010). Combining and Comparing Multiple Algorithms for Better Learning and Classification: A Case Study of MARF, Robot Learning, Suraiya Jabin (Ed.), ISBN: 978-953-307-104-6, InTech, Available from: http://www.intechopen.com/books/robot-learning/combining-and-comparing-multiple-algorithms-for-better-learning-and-classification-a-case-study-of-m

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821