Developing Pattern Recognition Systems Based on Markov Models: The ESMERALDA Framework[1]

G. A. Fink and T. Plötz

Intelligent Systems Group, Robotics Research Institute, Dortmund University of Technology
Otto-Hahn-Straße 8, 44227 Dortmund, Germany
e-mail: Gernot.Fink@udo.edu; Thomas.Ploetz@udo.edu

Abstract—In this paper we describe ESMERALDA—an integrated Environment for Statistical Model Estimation and Recognition on Arbitrary Linear Data Arrays—which is a framework for building statistical recognizers operating on sequential data as, e.g., speech, handwriting, or biological sequences. ESMERALDA primarily supports continuous density Hidden Markov Models (HMMs) of different topologies and with user-definable internal structure. Furthermore, the framework supports the incorporation of Markov chain models (realized as statistical *n*-gram models) for long-term sequential restrictions and Gaussian mixture models (GMMs) for general classification tasks. ESMERALDA is used by several academic and industrial institutions. It was successfully applied to a number of challenging recognition problems in the fields of automatic speech recognition, offline handwriting recognition, and protein sequence analysis. The software is open source and can be retrieved under the terms of the LGPL.

DOI: 10.1134/S1054661808020041

1. INTRODUCTION

The idea of statistical sequence models with finite memory first appeared almost a century ago, when the Russian mathematician Andrey Andreyevich Markov (1856–1922) applied such a type of model for the statistical analysis of character sequences in text [1]. The important algorithms for training and decoding so-called Hidden Markov Models (HMMs) were discovered around half a century later by Baum et al. [2] and Viterbi [3], respectively. At that time it was not yet clear whether a pattern recognition paradigm based on statistical models, the parameters of which can be learned from samples, would be successful in some domain, as the paradigm of classical artificial intelligence, which is based on declarative models and complex reasoning, was still predominant. The first successful pattern recognition system based on HMMs that became widely known was the HARPY speech understanding system [4]. Though the competition between rule-based and statistical methods was not yet decided at that time, the domain of automatic speech recognition became the field of research where HMM-based systems were further developed on a large scale. The models became known to a wider community of researchers from different areas by the famous tutorial paper of Rabiner [5]. Around that time it became also clear that, due to the availability of large speech databases collected within the DARPA programs, the statistical paradigm of speech recognition was superior to all rule-based approaches pursued so far. Only a few years later the vast majority of speech recognition systems were based upon a combination of HMMs for the statistical modeling of the realizations of acoustic events and Markov chain models for describing plausible sequences of words from a given lexicon—a situation which has not changed so far.

Compared to declarative or rule-based approaches for analyzing sequential data, the statistical paradigm has two important advantages. First, the vast majority of model parameters can be optimized on sample data. Second, the approach solves the problem that for a task requiring both segmentation and classification of the data neither step can be performed optimally in isolation. As Markov-model–based recognizers integrate segmentation and classification into one framework, the methods are also referred to as being *segmentation-free*.

From the other areas of pattern recognition that Markovian models began to conquer after their great success in the speech recognition field, the most important ones are the analysis of biological sequences and the recognition of machine printed or handwritten texts. However, the technology has been widely used to analyze sequential data of diverse types such as, for example, financial time-series or visually observed hand trajectories in gesture recognition.

Nowadays no one would doubt that Markov models—namely HMMs in combination with Markov chain models—have become the state-of-the-art tool when aiming at the analysis of data that is either explicitly time dependent as, e.g., speech signals or that can be linearized appropriately as, e.g., images of handwritten text lines.

[1] The text was submitted by the authors in English.

Received November 20, 2007

Though the core of the mathematical theory behind HMMs and Markov chain models is rather simple, it is still a challenging task to realize a successful application on the basis of Markov models for some domain. This is mainly due to the fact that in order to implement the relevant algorithms not only the basic theoretical but also quite some practical aspects have to be taken into account. One example is that the Viterbi algorithm for decoding HMMs, for which descriptions can be found in numerous places in the literature, in practice is not efficient enough for decoding realistic models and, therefore, needs to be enhanced by a suitable pruning strategy.

Therefore, the goal of the ESMERALDA framework is to provide researchers with the necessary methods and tools in order to be able to successfully develop HMM-based recognizers for "real-world" problems. In the implementation of the development environment, techniques were realized such that the important practical aspects were also addressed as, for example, efficiency in both training and decoding of the models, robustness of parameters estimated on limited sample sets, and the possibility to use HMMs and Markov-chain models in an integrated decoder.

In this paper an overview of the ESMERALDA system will be given including brief descriptions of applications realized so far using the ESMERALDA framework. In the following section, first the relevant concepts behind HMMs and Markov-chain models will be summarized. The overall architecture of ESMERALDA will then be described in Section 3, including a brief description of the modules available. Section 4 then gives an overview of the various applications realized so far using ESMERALDA.

2. MARKOV-MODEL CONCEPTS

For the analysis of sequential data, the use of Hidden Markov Models (HMMs) as statistical models has become the state of the art. In combination with statistical language models for describing restrictions of possible hypotheses sequences, powerful classification systems can be realized in numerous application domains. Generally, Markov models are applicable to all signal data evolving in time. Furthermore, when substituting time dependency with position or location dependency in a single dimension an even broader spectrum of data can be treated.

When applying Markovian models for pattern recognition purposes, one always assumes a statistical model for the generation of the data to be analyzed. A sequence of symbols or words $\mathbf{w}$ generated by some source is coded into a signal representation and later observed as a sequence of feature vectors $\mathbf{X}$. The goal of the recognition process then is to find the sequence $\hat{\mathbf{w}}$ that maximizes the posterior probability $P(\mathbf{w}|\mathbf{X})$ of the symbol sequence given the data.

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{X}) = \arg\max_{\mathbf{w}} \frac{P(\mathbf{w})P(\mathbf{X}|\mathbf{w})}{P(\mathbf{X})} = \arg\max_{\mathbf{w}} P(\mathbf{w})P(\mathbf{x}|\mathbf{w}). \quad (1)$$

When applying Bayes' rule, $P(\mathbf{w}|\mathbf{X})$ can be rewritten into a form where the two modeling components of Markov-model–based recognition systems become manifest. $P(\mathbf{w})$ denotes the language model probability for the sequence of symbols $\mathbf{w}$, and $P(\mathbf{X}|\mathbf{w})$ represents the probability of observing this sequence of symbols as sensor data $\mathbf{X}$ according to the "appearance" model, namely the HMM.

The fundamental advantage of Markov-model–based recognizers is that they do not require an explicit segmentation of the data prior to its classification. The recognition is thus performed in a *segmentation free* manner, which means that segmentation and classification are integrated.

In the following the theoretical principles of both hidden Markov and Markov-Chain models are briefly summarized including usage aspects and an overview of relevant algorithms. For a more thorough treatment of HMMs and n-gram models, the interested reader is referred to [6] and the references given therein.

2.1. Hidden Markov Models

Hidden Markov models describe a two-stage stochastic process with hidden states and observable outputs. The first stage represents a discrete stochastic process which produces a series of random variables that take on values from a discrete set of states. This process is *stationary*, which means that its statistical properties do not change over time, and also *causal* and *simple*. The last two properties taken together restrict the dependency of the probability distributions of states generated by the random variables to be dependent on the immediate predecessor state only. The Markov process is then said to be of first order:

$$P(s_t|s_1, s_2, \ldots, s_{t-1}) = P(s_t|s_{t-1}).$$

Basically, this first stage represents a finite state automaton which behaves probabilistically. In the second stage then at every time t an output is generated depending on the current state s_t only:

$$P(O_t|O_1 \ldots O_{t-1}, s_1 \ldots s_t) = P(O_t|s_t).$$

Since only these outputs O_t and not the associated internal states s_t can be observed, the overall model is referred to as *hidden* Markov model. Depending on the type of input data, the output elements generated per state can be either symbolic—i.e., of discrete type—or continuous. For pattern recognition purposes, the latter representation is suited better, as usually real-valued

vectors from some high-dimensional feature-space need to be processed. Consequently, the probability distributions of the statistical outputs of the model need to be able to define continuous distributions over $\mathbb{R}^N$. Since no general parametric families of such distributions are known, in the continuous case probability distributions are usually approximated via mixtures of Gaussians.

In summary, a *first order* hidden Markov model λ is formally defined as consisting of

• a finite set of states $\{s|1 \le s \le N\}$,

• a matrix of state transition probabilities[2] $\mathbf{A} = \{a_{ij}|a_{ij} = P(s_t = j|s_{t-1} = i)\}$,

• a vector of start probabilities $\pi = \{\pi_i|\pi_i = P(s_1 = i)\}$, and

• state-specific output probability distributions $\{b_j(O_t)|b_j(O_t) = p(O_t|s_t = j)\}$, which may be either discrete or continuous.

HMMs are attractive because there exist efficient algorithms for estimating the model parameters as well as for decoding the model on new data, which is equivalent to the aforementioned integrated segmentation and classification of the associated data.

For training the model, a variant of the well-known Expectation Maximization (EM) technique [7], namely the so-called Baum-Welch algorithm, is normally used. The method applies an iterative growth transformation to the model parameters such that the generation probability of the data given the model is improved:

$$P(\mathbf{O}|\hat{\lambda}) \ge P(\mathbf{O}|\lambda).$$

Here $\hat{\lambda}$ denotes the adapted HMM derived from the previous model λ by applying one reestimation step to the parameters.

The basis of model decoding is formed by the so-called Viterbi algorithm, which is used to—in the statistical sense—"infer" the hidden state sequence that with maximum probability generates an available sequence of outputs given the model:

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(\mathbf{O}, \mathbf{s}|\lambda).$$

As states can be associated with basic segmentation units, decoding yields the segmentation of the data considered on the basis of the current model.

The efficiency in both evaluating and decoding the model arises from the fact that HMMs store only one internal state as context for future actions, which is also called the Markov property. Therefore, computations necessary to obtain the production probability $P(\mathbf{O}|\lambda)$ and the optimal state sequence $\mathbf{s}^*$ can be performed in a dynamic programming style with linear complexity in the length of the sequence considered and quadratic complexity in the number of model states. Still, the algorithms are usually not efficient enough in practice, such that especially for decoding the model clever pruning strategies are applied.

[2] For practical applications the actual model topology—i.e., the connectivity between states of a certain model—is usually limited using specific, nonergodic model architectures (e.g., linear or Bakis type).

2.2. Markov Chain Models

In addition to the analysis of local context within sequential data, which is covered by HMMs, it is desirable in many applications to be able to describe long-term dependencies within the statistical modeling framework. In speech recognition, for example, where individual HMM states describe parts of elementary phonetic units, restrictions concerning the potential cooccurrences of subsequent words cannot be captured using HMMs alone. This is where Markov-chain models come into play.

Markov-chain models can be used to statistically describe the probability of the occurrence of entire symbol sequences. Formally speaking (cf. Eq. 1), the probability $P(\mathbf{w})$ of a sequence of symbols $\mathbf{w} = w_1, w_2, \ldots, w_T$ is calculated. In order to make things mathematically tractable, $P(\mathbf{w})$ is first factorized using Bayes' rule according to

$$P(\mathbf{w}) = P(w_1)P(w_2|w_1)\ldots P(w_T|w_1, \ldots, w_{T-1}) = \prod_{t=1}^{T} P(w_t|w_1, \ldots, w_{t-1}).$$

Since the context dependency increases arbitrarily with the length of the symbol sequence, in practice the "history" of a certain symbol is limited:

$$P(\mathbf{w}) \approx \prod_{t=1}^{T} P\underbrace{(w_t|w_{t-n+1}, \ldots, w_{t-1})}_{n \text{ symbols}}.$$

This means that the probability of the complete sequence is defined on the basis of the conditional probabilities of some symbol—or word—w_t occurring in the context of its $n-1$ predecessor words $w_{t-n+1}, \ldots, w_{t-1}$. Markov-chain models are, therefore, often referred to as n-gram or language models.

For the evaluation of n-gram models on unknown data, usually the perplexity $\mathcal{P}$

$$\mathcal{P}(\mathbf{w}) = \frac{1}{\sqrt[|w|]{P(\mathbf{w})}} = \frac{1}{\sqrt[T]{P(w_1, w_2, \ldots, w_T)}} = P(w_1, w_2, \ldots, w_T)^{-\frac{1}{T}}$$

is exploited as the evaluation criterion. Formally, the perplexity of some unseen data $\mathbf{w}$ is the cross-entropy between the symbol distribution defined by the probabilistic model and the one defined empirically by the data. The smaller the perplexity the better the n-gram model is able to predict the unseen data.

In principle, the conditional probabilities required for Markov-chain models can be derived from training data. However, even for moderate sizes of n (e.g., 2 for bigram models or 3 for trigram models) most n-gram events necessary for deriving robust statistical estimates will not be observed in a typical set of training data due to its limited size. Therefore, for robust estimation of n-gram models, it is of fundamental importance to appropriately smooth the raw probabilities in order to obtain useful probability estimates for events not observed in the training data (so-called unseen events). Therefore, in practical applications n-gram counts are modified and some "probability mass" for unseen events is gathered, e.g., by certain discounting techniques. The resulting zero-probability is then redistributed to unseen events according to a more general distribution. Widely used examples of smoothing techniques are Backing-Off and Interpolation (cf., e.g., [8]).

As HMMs and n-gram models are quite similar to each other, they can be combined rather easily into an integrated model (cf. Eq. 1). In order to balance between the different granularities of the models, a weighted combination of the different scores is necessary:

$$P(\mathbf{w})^{\rho} P(\mathbf{X}|\mathbf{w}).$$

Furthermore, as n-gram models span considerably longer contexts than HMMs, the search procedures used for integrated model decoding also become more complex.

3. SYSTEM OVERVIEW

In order to actually benefit from the capabilities of Markov models for recognition applications in real-world scenarios, implementations of the methods need to address not only the theoretical concepts outlined in the previous section. For building successful applications, important additional aspects have to be considered. The most important ones are numerical issues: the robust estimation of parameters on realistic, i.e., small, training sets; the actual application-specific configuration of the models; and the efficient model evaluation for interactive applications.[3] Those practical considerations primarily motivated the development of the toolkit described in this paper.

ESMERALDA—an integrated **E**nvironment for **S**tatistical **M**odel **E**stimation and **R**ecognition on **A**rbitrary **L**inear **D**ata **A**rrays—is a development environment for building statistical recognizers operating on sequential data such as, for example, speech, handwriting, or biological sequences. The framework primarily supports continuous density Hidden Markov Models of different topologies and with user-definable internal structure. Furthermore, Markov chain models (realized as statistical n-gram models), which can be used to complement HMM-based systems, are provided for representing long-term sequential restrictions. Additionally, Gaussian mixture models (GMMs), which are used to represent the output behavior of HMMs, can also be used in isolation for solving general classification tasks, where a model internal structure for representing local contextual restrictions is not required.

From the wealth of methods proposed in the context of Markovian models, it is the goal of ESMERALDA to put together a tractable set of conceptually simple yet powerful techniques in an integrated framework. The system consists of a modular architecture (cf. figure). Separate base-modules for estimating mixture density models (md) in conjunction with HMMs (mm) and for building n-gram language models (lm) are provided. Specialized algorithms necessary for certain application domains are provided in additional modules. These are the speech processing module (dsp), image processing module (im), handwriting recognition module (pen), and (biological) sequence processing module (seq).

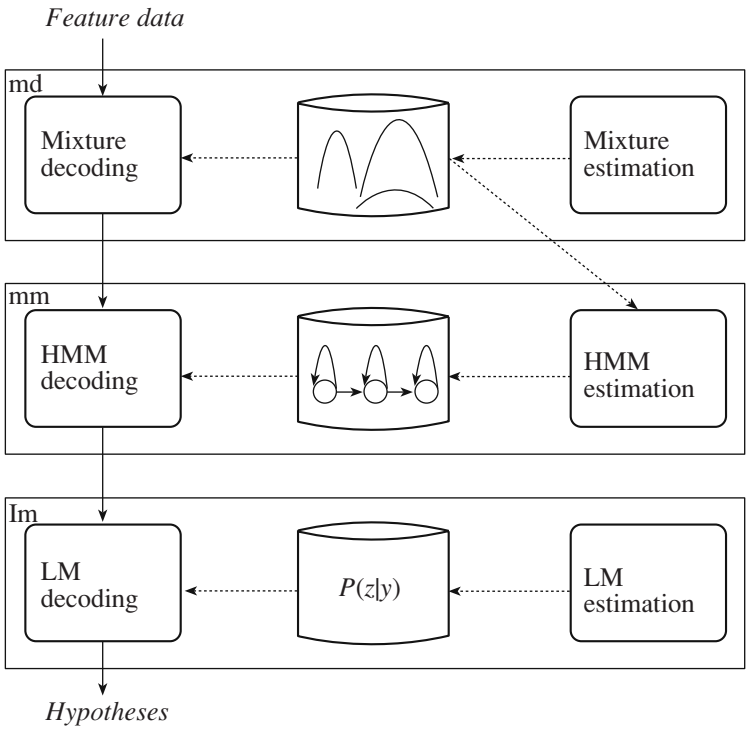
Furthermore, modules providing runtime system functionality (rs), fundamental linear algebra operations (mx), and tools for feature extraction and manipulation (fx), respectively, are the basis of the general framework. Technically, every module contains a library with an API as well as stand-alone programs for manipulating the appropriate models and associated data.

3.1. Mixture Densities

For describing the statistical output behavior of continuous HMMs that operate on streams of real-valued vectors from some high-dimensional feature space, mixtures of Gaussian distributions are widely used. The primary reason for this lies in the fact that these models are capable of approximating any general multimodal distribution over $\mathbb{R}^N$. ESMERALDA provides implementations of techniques for robust unsupervised mixture density estimation on API-level and as standalone programs, respectively. Standard clustering techniques, namely, k-means, Lloyd, and LBG, can be used for the initialization of mixture density models on unannotated data. The clustering process can be configured using various heuristics for robust parameter initialization. Following this, actual mixture model training based on Expectation Maximization can be performed using tools from ESMERALDA's md-module. In addition to mixture model estimation "from scratch," it might be favorable to actually adapt existing Gaussians exploiting domain specific training data. In this case maximum a-posteriori (MAP) adaptation of mixture density models can be performed using ESMERALDA.

By means of the md-module, certain linear feature space transforms, like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), can be estimated. Separate tools for estimating the transforms as well as for their application to the data are provided. In addition to this, for efficiency and for the sake of sav-

[3] A more detailed overview of practical issues relevant for Markov-model–based recognizers is given in [6, Part 2].



Overview of the system architecture of ESMERALDA. The development toolkit consists of three basic modules for estimating mixture density models (md) and HMMs (mm) and for building *n*-gram language models (lm). Additional modules (not shown here) provide tools for specific application domains and basic runtime system functionality (see text for details).

ing storage space, those transforms can also be directly integrated into the mixture density models allowing for "on-the-fly" calculations. Furthermore, for efficient model evaluation, a two-stage decoding scheme [9] can be used, which substantially reduces the computational effort required for mixture evaluation.

3.2. Hidden Markov Models

HMM-based recognizers estimated using ESMERALDA consist of elementary models specifying a certain topology, i.e., type of allowed state transitions. The framework provides built-in support for models with Linear, Bakis, Left–Right, Bounded Left–Right [10], and Profile topology. Furthermore, arbitrary model architectures can be defined by the user. Elementary models, independent of their particular topology, are built from individual state definitions that carry all statistical model parameters. From these models more complex and structured HMMs can be constructed by using a declarative specification language.

Model estimation is usually started with some kind of initialization (for model adaptation see below). ESMERALDA's mm-module provides automatic initialization procedures exploiting labeled data for HMMs based on elementary models with one of the built-in topologies. For actual model training, i.e., parameter optimization, the standard Baum-Welch reestimation algorithm (a variant of the EM-method) is implemented as stand-alone program within the mm-module. Furthermore, it can also be accessed at the API-level, i.e., as a library-function. For efficient parameter optimization, the Beam-search procedure is included in the training procedure.

For finding a good balance between precision of models and robustness of parameter estimates, ESMERALDA provides the possibility of state clustering. Based on an entropy criterion, similar states are grouped into clusters for which individual new parameter sets are created. These new parameters can then be optimized in subsequent reestimation steps. By means of ESMERALDA's parameter estimation capabilities as summarized here, robust and efficient HMM training can be performed with maximum exploitation of (limited) annotated sample sets.

When aiming at the adaptation of existing HMMs towards a specific target domain on limited sample data only, Maximum Likelihood Linear Regression (MLLR) as well as Maximum A-Posteriori (MAP) adaptation in recent years became state-of-the-art methods. ESMERALDA provides efficient implementations of those semisupervised adaptation techniques on API-level, and as stand-alone programs, respectively.

In order to evaluate Hidden Markov models on unknown data both the forward–backward algorithm and efficient Viterbi Beam-search decoding is provided by ESMERALDA's mm-module. In the latter the basic Viterbi algorithm is greatly enhanced in its efficiency—

especially when working with large scale models—by adding a pruning strategy that focuses the search to a "beam" of promising path hypotheses around the currently best solution (cf. [4]).

3.3. N-gram Models

Although Hidden Markov models are very powerful for various recognition tasks in many applications, their limited memory can be restricting w.r.t. recognition performance since long-term dependencies between sequential data cannot be covered. These dependencies might contain substantial discriminative information. Especially for large inventory recognizers, possible hypothesis sequences are desired to be restricted in some reasonable manner.

In order to cover long-term dependencies as mentioned before, the application of statistical language models can be considered state of the art. Since they can rather easily be combined with Hidden Markov models (cf. Eq. 1), Markov chain models—so-called *n*-gram models—are usually the methodology of choice for integrating long-term restrictions into statistical recognition systems.

Therefore, ESMERALDA's lm-module includes tools for estimating arbitrary *n*-gram models. Methods for redistributing probability mass and smoothing distributions (in order to assign robust nonzero probability estimates to unseen *n*-grams) are used. More specifically, ESMERALDA allows for memory efficient estimation of *n*-gram statistics analyzing ordinary text data. Based on these statistics, *n*-gram models can be estimated efficiently based on different smoothing techniques. Most notably, absolute and linear discounting are implemented as well as backing-off and interpolation. Furthermore, tools are provided allowing for an efficient decoding of long-span models.

N-gram related techniques are accessible at API-level, i.e., as library functions, and as standalone programs, respectively, allowing for the rapid development of user-defined recognition systems. Furthermore, a ready-to-use integrated recognizer combining both HMMs and statistical language models is provided with the toolkit.

3.4. Integrated Development Environment

ESMERALDA was designed with special focus on the development of recognition systems which can be embedded into "real-world" applications. Therefore, a command line interface has been developed allowing for pipelined operation by cascading the particular modules. All external representations of ESMERALDA-related data (e.g., HMM parameters, mixture density models, etc.) come as human-readable ASCII-data allowing for easy reproducibility and debugging.[4] Besides the three core-modules described in the previous sections (namely md, mm, and lm), certain additional modules provide runtime system and application specific functionality, respectively. In the following a brief overview of those modules including the algorithms implemented will be given.

Runtime System (rs): The purpose of ESMERALDA's runtime system is to provide fundamental system handling functions which make the software creation process easier and more convenient for developers. The rs-module consists of functions addressing basic input/output operations, memory management, and access to operating system features (like timers etc.). Furthermore, basic data types and related algorithms are included.

Linear Algebra (mx): For most pattern recognition systems, especially for those based on Markov models, linear algebra methods represent the core of related algorithms. It is the goal of ESMERALDA's mx-module to provide a comprehensive developer's toolkit including such basic linear algebra methods. Examples are basic matrix operations (e.g., multiplication, summation, transpose, etc.) or more complex operations like inversion, efficient and robust calculation of Eigenvalues, or solving linear equation systems.

Feature Extraction and Manipulation (fx): Pattern recognition systems are usually based on (high-dimensional) real-valued feature vectors. For certain basic manipulations like computing derivatives, extraction of components, merging, smoothing, calculation of statistics, and (linear) transformations, ESMERALDA's fx-module provides various tools. These stand-alone programs can easily be integrated into high-throughput workflows, e.g., for transforming complete sample sets.

Evaluation (ev): The evaluation of classification experiments based on the results provided by the particular recognizer can be tedious work. For convenience the ev-module provides tools which allow for the computation of error rates for classification, segmentation, and detection experiments (e.g., for tasks where certain patterns need to be localized within large sequences). In the latter case, ROC and DET curves can be created based on numerous different evaluation criteria. For easy visualization an interface to gnuplot is included.

ESMERALDA also includes modules that provide specialized functionality required for certain application domains.

Speech Processing (dsp): One of the many application areas of ESMERALDA was and still is automatic speech recognition. The dsp-module implements the required preprocessing and feature extraction techniques. Algorithms for the computation of mel-frequency cepstral coefficients (MFCCs) including causal cepstral mean-normalization are available. Additionally, the dsp-module provides basic methods for voice activity detection.

[4] Feature data, however, is the only (reasonable) exception which is stored in (raw) binary ANSI/IEEE 854 single precision floating-point format.

Image Processing (im): Especially for recognition systems addressing the analysis of image data, ESMERALDA's im-module is provided. It includes easy-to-use generalized image data types and certain standard (filtering) algorithms (e.g., Canny, Sobel, etc.). Currently, automatic handwriting recognition represents the most prominent application of Markov-model–based image analysis using ESMERALDA and its im-module (cf. Section 4.2).

Note that the im-module is not restricted to use in Markov-model–based recognition systems only. In fact it represents a more general, easy-to-use image processing toolbox which already has been successfully applied especially in batchlike workflows.

Handwriting Recognition (pen): Using the core modules of ESMERALDA and the aforementioned im-module, the pen-module provides certain high-level, ready-to-use tools required for setting up complete (offline, i.e., image based) handwriting recognition systems. This includes image preprocessing as well as feature extraction necessary prior to the actual HMM- and n-gram–based modeling and recognition, respectively.

(Biological) Sequence Analysis (seq): The analysis of biological sequences requires certain specialties in treatment of the data at the technical level. As an example the different inventories specific to particular types of data need to be treated. ESMERALDA supports all kind of symbolic input data by allowing for direct mapping of the particular symbols to (binary) vectors. Thus, e.g., nucleotides of DNA sequences are mapped to four-dimensional vectors with Adenin represented as $(1000)^T$, Cytosin as $(0100)^T$, and so forth.

Additionally the seq-module contains certain tools enabling batch processing workflows for automated high-throughput analysis of biological sequences on a large scale.

The framework is completely written in ANSI-C. Currently, it runs on several UNIX-like operating systems (including Linux, MacOS X, and Solaris). Due to its modular architecture and the convenient API, it can easily be used and extended for various applications domains.

The software is open source and can be retrieved either from the authors directly or from sourceforge[5] under the terms of the GNU Lesser General Public License (LPGL).

4. APPLICATIONS

The development of ESMERALDA started more than a decade ago and the framework meanwhile reached a rather mature state.[6] Various researchers are now actively using it for their own work in academia. Furthermore, the toolkit has also been used in several industrial applications serving as a backbone recognition framework in rather complex systems addressing the analysis of sequential data.

As specific examples, ESMERALDA has already successfully been applied to a number of challenging pattern recognition problems in the field of automatic speech recognition, offline handwriting recognition, protein sequence analysis, and analysis of music data. In the following a brief overview of ESMERALDA's application within these domains is given.

4.1. Speech Recognition

Originally designed for speech recognition purposes, the use of ESMERALDA within this domain has a fairly long history. Allowing for batch as well as for interactive speech recognition, signal recording and feature extraction (based on MFCCs) modules are integrated.

Within an incremental speech recognizer, all calculations from feature extraction to language model search are carried out strictly time-synchronously [12]. In order to be able to produce recognition results for an utterance while the user is still speaking, i.e., the end of the input signal is not yet reached, an incremental processing strategy was developed. Additionally the recognizer is capable of applying the constraints of a context-free grammar in conjunction with a statistical language model [13]. In [14] acoustic and articulatory information have been combined using ESMERALDA for robust speech recognition.

As prominent examples, the toolkit has been used for the development of online speech recognizers embedded in intelligent human–robot interaction systems (cf., e.g., [15, 16]) including automatic speaker identification [17]. Furthermore, a recognizer for accessing nonsafety relevant functions of cars was realized [18] including online adaptation to changing acoustic environments [19].

4.2. Handwriting Recognition

The use of HMM-based techniques for handwriting recognition was inspired by the success of statistical models in the field of automatic speech recognition [20]. Over the last decade, those models became popular and were applied with great success by a number of research groups. Depending on the recording process, handwritten script is either processed as online (i.e., trajectories of pen movements captured by pressure sensitive tablets) or offline data (i.e., digital document images acquired by, e.g., scanners or video cameras). In the latter case, lines of script are extracted from the images of the handwriting data and are usually subject to several preprocessing and normalization operations. Subsequently, sequential data is extracted, by means of a sliding window approach resulting in a stream of features.

In recent years ESMERALDA has been successfully used for realizing offline handwriting recognition systems (cf., e.g., [21]). Feature streams are calculated

[5] http://sourceforge.net/projects/esmeralda.

[6] An early description of (parts of) the toolkit in the context of automatic speech recognition can be found in [11].

from lines of handwritten script which are automatically extracted from the particular documents analyzed. By means of ESMERALDA, HMMs with Bakis topology are estimated for letters which are combined to word models using the framework's configuration language. The integration of bigram models restricts the decoding reasonably.

ESMERALDA has also been used for the realization of a video-based whiteboard reading system which recognizes handwritten notes [21].

4.3. Biological Sequence Analysis

The functions of proteins, which are of major interest for life-science applications, are more or less directly connected to their primary structure, i.e., the underlying sequence of amino acids. Due to the linear structure of this biological data, (complex) Profile HMMs with specific topologies have been applied very successfully to genomics and proteomics tasks [22].

By means of the ESMERALDA framework, substantial enhancements of the basic approach to sequence alignment have been developed (cf., e.g., [10, 23]) improving the detection of remotely related protein sequences, which is especially relevant for pharmaceutical purposes. Therefore, a new feature representation for protein sequences was developed [24] allowing for semicontinuous protein family HMMs with less complex model architectures [25].

5. SUMMARY

In this paper we presented the ESMERALDA framework—an integrated Environment for Statistical Model Estimation and Recognition on Arbitrary Linear Data Arrays. It provides a conceptually simple yet powerful set of methods and tools for building pattern recognition systems based on Markov models. It can be applied to solve challenging recognition problems on sequential data as, for example, speech, handwriting, or biological sequences. ESMERALDA primarily supports continuous density Hidden Markov Models (HMMs). Complex HMMs can be created from elementary models, for which different topologies or a completely user-definable internal structure can be specified, by using a declarative configuration language. The framework provides the relevant algorithms for parameter estimation and model decoding. In addition to HMMs for capturing statistics of local contexts in data ESMERALDA also supports the creation and use of statistical *n*-gram models (i.e., Markov chain models), which can be used to describe long-term sequential restrictions. As these models are compatible with HMMs, they can be used in a combined fashion for integrated decoding. The Gaussian mixture models, which are part of continuous HMMs for describing their output behavior, can also be used independently with the tools and methods provided by ESMERALDA for solving standard classification tasks where no internal model structure is necessary.

Currently, ESMERALDA is used by a number of institutions in academia and industry for research purposes and for developing "real-world" applications. Several challenging recognition problems were successfully tackled by using the ESMERALDA framework. The most important domains investigated are automatic speech recognition, offline handwriting recognition, and the detection of remotely homologous protein sequences in large databases.

The software is open source and can be retrieved from sourceforge[7] under the terms of the GNU Lesser General Public License (LPGL).

ACKNOWLEDGMENTS

The authors would like to acknowledge the substantial support for the development of ESMERALDA provided by their former affiliation, namely the Applied Computer Science Group of Bielefeld University, Bielefeld, Germany, and the significant contributions to the image processing module by Marc Hanheide and Frank Lömker.

REFERENCES

1. A. A. Markov, "Example of Statistical Investigations of the Text of "Eugen Onegin," which Demonstrates the Connection of Events in a Chain," in *Bulletin de l'Académie Impériale des Sciences de St.-Pétersbourg* (Sankt-Petersburg, 1913), pp. 153–162 [in Russian].
2. L. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Ann. Math. Statist. **41**, pp. 164–171 (1970).
3. A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. on Information Theory **13**, 260–269 (1967).
4. B. Lowerre and D. Reddy, "The Harpy Speech Understanding System," in *Trends in Speech Recognition*, Ed. by W. Lea (Englewood Cliffs, Prentice-Hall Inc., New Jersey, 1980), pp. 340–360.
5. L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," in *Proceedings of the IEEE, 1989*, vol. 77, no. 2, pp. 257–286.
6. G. A. Fink, *Markov Models for Pattern Recognition, From Theory to Applications* (Springer Heidelberg, 2008).
7. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society, Series B **39** (1), 1–22 (1977).
8. S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Speech and Language **13**, 359–394 (1999).
9. E. G. Schukat-Talamazzini, M. Bielecki, H. Niemann, T. Kuhn, and S. Rieck, "A Non-Metrical Space Search Algorithm for Fast Gaussian Vector Quantization," in

[7] http://sourceforge.net/projects/esmeralda.

Proceedings Int. Conf. on Acoustics, Speech, and Signal Processing (Minneapolis, 1993), pp. 688–691.

10. T. Plötz and G. A. Fink, "Pattern Recognition Methods for Advanced Stochastic Protein Sequence Analysis Using HMMs," Pattern Recognition, Special Issue on Bioinformatics **39**, 2267–2280 (2006).
11. G. A. Fink, "Developing HMM-Based Recognizers with ESMERALDA," in *Lecture Notes in Artificial Intelligence*, Ed. by V. Matoušek, P. Mautner, J. Ocelíková, and P. Sojka (Springer, Berlin Heidelberg, 1999), vol. 1692, pp. 229–234.
12. G. A. Fink, C. Schillo, F. Kummert, and G. Sagerer, "Incremental Speech Recognition for Multimodal Interfaces," in *Proceedings Annual Conference of the IEEE Industrial Electronics Society, 1998*, vol. 4, pp. 2012–2017.
13. S. Wachsmuth, G. A. Fink, and G. Sagerer, "Integration of Parsing and Incremental Speech Recognition," in *Proceedings European Signal Processing Conference, 1998*, vol. 1, pp. 371–375.
14. K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining Acoustic and Articulatory Information for Robust Speech Recognition," Speech Communication **37** (3–4), 303–319 (2002).
15. A. Haasch, S. Hohenner, S. Hüwel, M. Kleinehagenbrock, S. Lang, I. Toptsis, G. A. Fink, J. Fritsch, B. Wrede, and G. Sagerer, "BIRON—The Bielefeld Robot Companion," in *Proceedings Int. Workshop on Advances in Service Robotics,* Ed. by E. Prassler, G. Lawitzky, P. Fiorini, and M. Hägele (Fraunhofer IRB Verlag, Stuttgart, Germany, May 2004), pp. 27–32.
16. G. A. Fink, J. Fritsch, N. Leβmann, H. Ritter, G. Sagerer, J. J. Steil, and I. Wachsmuth, "Architectures of Situated Communicators: From Perception to Cognition to Learning," in *Situated Communication,* Ed. by G. Rickheit and I. Wachsmuth, pp. 357–376 (Berlin, Mouton de Gruyter; Trends in Linguistics, 2006).
17. G. A. Fink and T. Plötz, "Integrating Speaker Identification and Learning with Adaptive Speech Recognition," in *2004: A Speaker Odyssey—The Speaker and, Language Recognition Workshop*, pp. 185–192 (2004).
18. C. Schillo, G. A. Fink, and F. Kummert, "Grapheme Based Speech Recognition for Large Vocabularies," in *International Conference on Spoken Language Processing* vol. 4. (Beijing, China, 2000), pp. 584–587.
19. T. Plötz and G. A. Fink, "Robust Time-Synchronous Environmental Adaptation for Continuous Speech Recognition Systems," in *International Conference on Spoken Language Processing* (2002).
20. T. Starner, J. Makhoul, R. Schwartz, and G. Chou, "Online Cursive Handwriting Recognition Using Speech Recognition Methods," in *Proceedings Int. Conf. on Acoustics, Speech, and Signal Processing, 1994*, vol. 5, pp. 125–128.
21. M. Wienecke, G. A. Fink, and G. Sagerer, "Toward Automatic Video-Based Whiteboard Reading," Int. Journal on Document Analysis and Recognition **7** (2–3), 188–200 (2005).
22. R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (Cambridge University Press, 1998).
23. T. Plötz and G. A. Fink, "Robust Remote Homology Detection by Feature Based Profile Hidden Markov Models," Statistical Applications in Genetics and Molecular Biology **4** (1) (2005).
24. T. Plötz and G. A. Fink, "Feature Extraction for Improved Profile HMM Based Biological Sequence Analysis," in *Proceedings Int. Conf. on Pattern Recognition* (IEEE, 2004), no. 2, pp. 315–318.
25. T. Plötz and G. A. Fink, "A New Approach for HMM Based Protein Sequence Modeling and Its Application to Remote Homology Classification," in *Proceedings Workshop Statistical Signal Processing* (IEEE, Bordeaux, France, 2005).

Gernot A. Fink received a diploma in computer science from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1991 and the Ph.D. degree (Dr.-Ing.) also in computer science from Bielefeld University, Germany, in 1995. In 2002 he received the venia legendi (Habilitation) in applied computer science from the Faculty of Technology of Bielefeld University.

From 1991 to 2005 he was with the Applied Computer Science Group at the Faculty of Technology of Bielefeld University. Since 2005 he is professor for Pattern Recognition in Embedded Systems at Dortmund University of Technology, Germany, where he also heads the Intelligent Systems group at the Robotics Research Institute (IRF). His research interests lie in the development and application of pattern recognition methods in the fields of man machine interaction, multimodal machine perception including speech and image processing, statistical pattern recognition, handwriting recognition, and the analysis of genomic data.

Dr. Fink is Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Signal Processing Society, and the IEEE Computer Society.

Thomas Plötz received a diploma in technical computer science from the University of Cooperative Education Mosbach, Germany, in 1998. He received a diploma and the PhD degree (Dr.-Ing.) in computer science from the University of Bielefeld, Germany, in 2001 and 2005, respectively.

From 2001 to 2006 he was with the Applied Computer Science Group at the Faculty of Technology of Bielefeld University. In 2006 he joined the Intelligent Systems group at the Robotics Research Institute of Dortmund University of Technology, Germany, where he holds a senior research position.

Dr. Plötz is interested in general aspects of machine learning and pattern recognition techniques and applications for various domains like speech-processing, automatic recognition of handwritten script, image processing, or bioinformatics. He is coordinating various research activities within the smart environment project at the Robotics Research Institute's "Intelligent House"—the FINCA.