Rethinking Algorithm Design and Development in Speech Processing

Thilo Stadelmann*, Yinghui Wang*, Matthew Smith†, Ralph Ewerth*, and Bernd Freisleben*
*Department of Math. and Computer Science, University of Marburg, Germany
{stadelmann, wangy, ewerth, freisleb}@informatik.uni-marburg.de
†Department of El. Engineering and Computer Science, University of Hannover, Germany
smith@rvs.uni-hannover.de

Abstract

Speech processing is typically based on a set of complex algorithms requiring many parameters to be specified. When parts of the speech processing chain do not behave as expected, trial and error is often the only way to investigate the reasons. In this paper, we present a research methodology to analyze unexpected algorithmic behavior by making (intermediate) results of the speech processing chain perceivable and intuitively comprehensible by humans. The workflow of the process is explicated using a real-world example leading to considerable improvements in speaker clustering. The described methodology is supported by a software toolbox available for download.

1 Introduction

Contemporary speech processing systems are complex, typically consisting of several algorithms. These often contain sub-algorithms, with numerous processing steps whose effects and parameter settings are not intuitively understandable by humans. This leads to several problems when designing new and adapting or replicating existing algorithms. Taking the mel frequency cepstral coefficients (MFCC) algorithm for speech features as a concrete example, parameters such as the number of coefficients to keep are relatively easy to understand, but other parameters, such as the window type or the size of the filter bank, are more abstract, making it difficult to intuitively judge their importance and their effects on the complete processing chain.

When adapting an existing algorithm to a new environment, there is usually no instant success due to such misconceptions. The same is true for designing a new algorithm based on theoretical results or reimplementing a published algorithmic description for comparison. When the results do not meet the expectations, several questions arise: What effect does a change of a parameter in a component of an algorithm have? What does the selection of a particular algorithmic technique in the presence of several possibilities have on the overall functionality? What is the contribution of a specific algorithmic step? Is it actually the right algorithm for this data? If not, how should a valid one be designed?

These questions are aimed at finding a hypothesis– the beginning of the scientific process. But how to arrive at a promising hypothesis? Some disciplines have developed their own methodologies to assist human creativity in this process. They conceptualize a principle that in its core is as appealing as common sense, then add to it formal procedures and ready-to-use tools. One such methodology, from the discipline of data mining, can be summarized by the phrase "know your data": the approach of striving for (visual, mathematical, expertise-like) insight into the data set belongs to every data miner's toolbox, making the mining process more amenable to planning and success more likely.

In this paper, we conceptualize a related methodology for speech processing that systemizes the search for hypotheses about the reasons of unexpected algorithmic behavior. The core principle and its relevance to the speech processing community is discussed in Section 2. Section 3 then formalizes the method by proposing a concrete workflow and provides tools. In Section 4, the method is then applied to a real world example from the area of speaker clustering. Section 5 concludes the paper and outlines areas for future work.

2 Problem Refinement

Our aim is to propose a method that helps making reasons for failure in complex (compositions of) speech processing algorithms graspable by humans. Grasping

This work is funded by the Deutsche Forschungsgemeinschaft (SFB/FK615, Project MT).

contains a certain extent of intuition. If an issue is intuitively clear, human creativity may generate hypotheses. Thus, stated informally, *seeking intuition* is the core of our approach. Obviously, most researchers strive for intuition in order to make discoveries. But how can intuition be achieved?

For researchers in the field of computer vision it is particularly easy to gain intuitive understanding using visualization, since their objects (and, often, results) of analysis are original visual objects. Arguably, this makes the visual domain a good choice to transform data into, in order to grasp their meaning. The same is true for the data mining area, where visualization is often applied to comprehend neighborhood relationships, a task that humans naturally associate with visual representations [3]. However, the success of visualization methods and the corresponding reliance of researchers on them can also be a hindering factor in other areas of research, because visualization is not in itself the only mediator of intuition. It is one of the possible transformations applicable to the data in order to find a representation for which we as humans are experts in perceiving meaning due to our natural abilities.

For example, in speech processing, the original domain of the input data is the auditory perception. There are still many applications for visualization in speech processing, but representing the speech signal's most prominent features as an image (the single popular technique here is the spectrogram) does not result in more intuition, but creates a higher-dimensional signal that needs an expert interpreter to make use of its many merits [6]. In the worst case, mere visualization transforms the data into an unnatural domain, thereby implicitly reducing the range of understandable or discoverable features to what the transformation can and cannot do. If the way of visualization is not suitable for a given problem, researchers may–devoid of knowing alternative ways–refrain from seeking intuition altogether, thereby risking to miss discoveries.

Mere visualization is not enough to let intuition emerge. For this purpose, we need to recast algorithmic sub-results to the specific perceptual domain in which we as humans are experts in intuitively grasping the context, the character and the reasons of the issue at hand. This subsumes visualization, but broadens the view to other possible transformations like resynthesis ("audibilization") by expecting insight not from an image alone, but from the unison of a domain suitable for the data *and* natural human grasping. We need other methods to achieve intuition, and particularly in speech processing there is a need for new developments, as Hill remarks [2]: the area currently misses a culture of perceptually motivated research, partly induced by missing methodologies and tools.

The contribution of this paper is threefold: first, it motivates the use of intuitive methods in the design and development of speech processing algorithms by presenting arguments and a successful example. Second, it facilitates the use of intuitive methods beyond visualization by proposing a methodology and workflow. This includes prerequisites and steps to follow on the way to hypothesizing solutions to the questions raised in the introduction. Third, it enables the use of intuitive methods by making available accompanying tools for multimodal intuitive analysis on the web.

3 Proposed Methodology and Workflow

We propose the following methodology to strive for intuition about the reasons of unexpected algorithmic outcomes: The starting point is an *existing algorithm* (or a process consisting of several algorithms) along with a certain *problem*, i.e. a question to- or aspect of interest in the algorithm. The problem might be as general as an observed malfunctioning (for example, a change detection algorithm operating at an unacceptable error rate) or as concrete as needing a good parameter setting.

The initial step is to identify all important phases in the algorithm or process. These phases do have intermediate results as implicit outcomes (the *data*). We seek insight into the algorithmic phase by perceptually observing its produced data, thereby feeling what has changed since the previous phase and whether the action has worked reasonably. Therefore, it is necessary to transform the sub-results into a *suitable domain*.

The suitable domain is a specific *gestalt* into which the data is transformed: for example, not just a sound, but male speech or single-tone music; not just an image, but a histogram or a gray scale gradient map. The suitable domain is characterized by the following property: it represents the data through metaphors humans use so frequently in everyday life that they judge their meaning rather implicitly (intuitively) than explicitly (rationally). This makes the suitable domain dependent on the problem, the data and the observer. An example for such a metaphor and corresponding intuitive judgment might be a red light in the traffic sign domain, regarding the question whether to continue a certain action; or a male voice in the speech domain regarding the question of the speaker's gender: a human observer knows the answer to the initial question after such a transformation without reflection. This instant awareness of either the answer to the initial problem, or other perplexing facts leading to new ways of thinking about the problem, is a frequent property of the presented approach.

Being aware of the need of- and subsequently find-

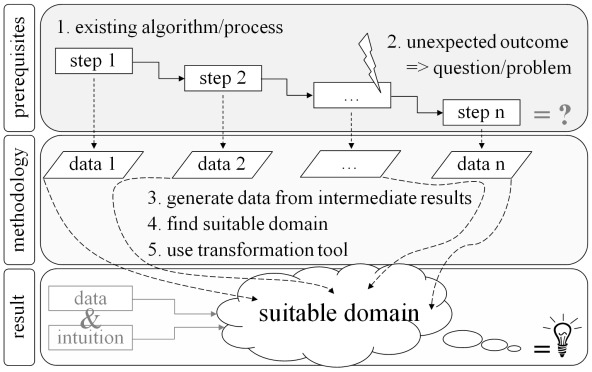

Figure 1. The proposed workflow.

ing a suitable domain for the transformation is the most important part of the workflow. It is in itself a creative process that can not be fully automated. Still, with the following aids, it is easier to pursue this search than to generate hypotheses about the initial problem deaf-blindly: first, empirically, a suitable domain is often the one that corresponds naturally with the investigator's imagination of the data under the given problem. For example, one may think of feature distributions as mountain massifs. Second, if the data represents an object of the physical world rather than some abstract intermediate result, the domain of this physical counterpart gives useful insights about a suitable domain for the transformation–possibly, the transformation consists of undoing previous transformations, as resynthesis does in speech processing. For instance, a statistical voice model can be imagined as something that really sounds like a voice without being comprehensible. Both examples, carried out as transformations on speech data, were helpful to solve certain problems in the past [8][7].

After a suitable domain is identified, the last step is to find or design a *tool* that carries out the transformation. Tools for this purpose will not be completely generic. However, a comprehensive archive of resynthesis- and visualization tools for most purposes in speech processing has been compiled on our website[1], together with source code, examples and other resources. It comprises, among others, software to make most common speech features and -models audible and new tools to visualize Gaussian mixture-based models.

Stepping through this workflow as shown in Figure 1 leads to a vivid representation in a suitable domain, allowing an experience of the inner workings of the algorithm under consideration. This provides a breeding ground for hypotheses about their failure in the given context. Revisiting the above-mentioned examples, if the mountain massif is too spiky, this may indicate changing the smoothness parameters of the distribution estimation technique, as is commonly understood. If the resynthesized voice model sounds not at all like a voice, this provokes further inquiries about possibly missing features in the data. The prerequisite for the methodology to work is that there is a representation of the data that corresponds with intuition. Fortunately, many patterns in speech processing have a natural origin and many pattern recognition problems a corresponding real world task they refer to.

The methodology described provides a framework for discovering reasons and possible solutions for problems in existing algorithms. This is useful for researchers when working on, adapting or extending present algorithms as well as for practitioners in debugging complex systems. But the practical relevance goes further: intuitive insight into state-of-the-art methods also makes their possible flaws and oversimplifications obvious. This can inspire completely new algorithms in an explorative way, thereby becoming a method of algorithm design rather than pure analysis. A third application is teaching: making algorithmic steps perceivable adds intuition and practical experience to theoretical understanding, conveying a keen sense for applications.

4 A Case Study

Next, we apply the proposed methodology step by step to a recent problem in speaker clustering. The focus here is on *how* the results have been achieved in order to exemplify the workflow, not on the results themselves; more details are available in another paper [7].

MFCC feature vectors modeled by Gaussian mixture models (GMM) are commonly used for the task of speaker identification, but also for the more complex task of speaker clustering. The final error rate is higher in a clustering experiment than for an identification task [4] under otherwise identical circumstances. Arguably, the used techniques are not expressive enough for the additional degrees of freedom introduced in clustering, i.e. they fail to represent something that becomes more important as soon as the task gets more difficult. So, how can clustering be improved for tasks depending on it, such as person-based video retrieval? Our testbed is adopted from Reynolds [5] as a clear configuration for evaluating MFCCs and GMMs for speaker recognition on the TIMIT database. Instead of identifying the speaker of each of the 630 test utterances as one of the 630 pre-built speaker models, we confine the database to 40 test- and 40 training utterances and perform a clustering of these 80 utterances.

To pursue the question raised above and to start with

[1] http://www.informatik.uni-marburg.de/~stadelmann/eidetic.html.

the workflow, we define the *algorithm* under consideration: the complete processing chain of speaker clustering. The chain can be partitioned into feature extraction (everything until valid MFCCs exist), model building (GMM training) and clustering (model comparison and unification). Because the success of the last phase depends largely on the quality of the voice models, we omit it from further analysis, keeping the phases of MFCC feature extraction and GMM model building.

Next, the *problem* needs containment. The speaker clustering chain is large enough for improvements at the wrong point not being able to propagate until its end. Thus, first, we need to find the bottleneck in the two identified phases. Second, we need a qualitative statement on what exactly is missing at this bottleneck.

The *data*, i.e. the intermediate results of the two phases, are two representations of a voice: the feature extraction yields a matrix of MFCC feature vectors. The model building process yields the parameter vectors of a GMM that represent the statistical properties of the vector set and, hopefully, of the voice. This suggests a *suitable domain* for the transformation: if we could listen to what is contained in the features and models, missing information may be easily identified.

Using this information, we design a respective *tool* to perform the necessary resynthesis. Listening yields a surprising result: resynthesized GMM voice models sound extremely strange to human ears, due to the resulting audio frames being completely independent of each other. This does not allow the emergence of intonation and hence creates no sensation of listening to speech. A user study has confirmed the first suspicion: as far as human investigators under this transformation are concerned, the missing time coherence information in the voice models is the desired bottleneck.

The insights gained by applying the proposed methodology and workflow inspire a proof of concept implementation: still using MFCCs as feature vectors (accompanied by pitch information), time coherence information can be implemented into density-based voice models by modeling context vectors. A context vector results from the concatenation of several subsequent MFCCs, so that the complete length of the vector roughly spans a syllable. Due to the higher dimensionality of context vectors, we exchange the GMM with a one-class support vector machine and achieve a 56.66% reduction of the diarization error rate in the given clustering experiment – from 4.53% to 1.96%.

5 Conclusions

In this paper, we presented a methodology to generate hypotheses about why algorithms in speech processing do not behave as expected. This human-in-the-loop approach strives for intuition into the problems by transforming algorithmic (sub-)results to a domain of perception where the human mind is considered to be an expert in conceiving the context and meaning of events, features and models naturally. We summarize this idea by the phrase "eidetic design" as in "eidetic reduction" of phenomenology: it describes a method by which the researcher achieves intuition into the pure essence of an issue apart from what blurs its image [1].

Using the workflow introduced in this paper, it has practical applications in algorithm design, development and debugging as well as in teaching. The methodology emerged from our own experience in researching and implementing speech processing systems and has shown its effectiveness several times. We exemplified the methodical process by applying it step by step to one of our real world examples, leading to profound algorithmic improvement. More examples, resources and tools are available on our website.

Since eidetic design depends on tools that impart the inner workings of an algorithm, future work will include developing new tools for other speech processing problems. Furthermore, we plan to apply the method to other fields, such as general multimedia analysis applications. Finally, we see promising first results on computer security-related algorithms.

References

[1] Encyclopædia Britannica. Eidetic reduction (philosophy). Online: `www.britannica.com/EBchecked/topic/180957/eidetic-reduction`, 05. Jan. 2010.

[2] D. R. Hill. Speaker Classification Concepts: Past, Present and Future. In C. Müller, editor, *Speaker Classification I – Fundamentals, Features, and Methods*, volume 4343 of *LNAI*, chapter 2, pages 21–46. Springer, 2007.

[3] D. A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):100–107, January–March 2002.

[4] M. Kotti, V. Moschou, and C. Kotropoulos. Speaker Segmentation and Clustering. *Signal Processing*, 88:1091–1124, 2008.

[5] D. A. Reynolds. Speaker Identification and Verification using Gaussian Mixture Speaker Models. *Speech Communication*, 17:91–108, 1995.

[6] P. Rose. *Forensic Speaker Identification*. Taylor & Francis, London and New York, 2002.

[7] T. Stadelmann and B. Freisleben. Unfolding Speaker Clustering Potential: A Biomimetic Approach. In *Proceedings of ACM Multimedia 2009*, pages 185–194, Beijing, China, October 2009.

[8] T. Stadelmann and B. Freisleben. Dimension-Decoupled Gaussian Mixture Model for Short Utterance Speaker Recognition. In *Proceedings of ICRP'2010*, Istanbul, Turkey, September 2010. To appear.