Ensemble-based Human Communication Recognition

P. Barthelmess

Department of Computer Science
University of Colorado at Boulder

Abstract. We propose a novel architecture for systems that target the recognition of human communication - *Distributed Ensembles*. *Distributed Ensembles* results from the observation that in many different fields hard problems are handled by employing multiple computational entities that cooperate to solve a problem. Even though these solutions share this common trait, the goals in each field for employing multiple computational entities can be very different from each other, and can be as distinct as reducing error-rates in Automatic Speech Recognition, obtaining faster convergence in optimization problems, and achieving high performance within multimodal recognition systems through parallel algorithms. As a consequence of the variety of goals, the structure of the entities and the nature of cooperation are also varied and it is usually the case that solutions do not reap the full benefits that we see could potentially be obtained. While existing solutions emphasize different aspects, our observation is that these aspects are not mutually exclusive, but complementary. Three of these aspects seem particularly useful in a general sense: performance and scalability through multiprocessing; faster convergence by sharing partial results; error reduction by combination of hypotheses. We therefore propose a style that has potential for combining these three aspects, based on ensembles of distributed computational entities. To enable the use of the envisioned approach, we propose to develop an architectural infrastructure to offer efficient coordination services that are exposed as an API to facilitate use from a developer's perspective. We explore issues surrounding Distributed Ensembles in the context of automatic recognition of human communication, that we show offer a particularly promising field of application. More specifically, we focus on American Sign Language (ASL) Recognition, a problem amenable to well researched natural language processing approaches.

1 Introduction

1.1 Cross-cutting ensemble-based solutions

A class of solutions that employ multiple Computational Entities (CE) that work at the same time on the same problem can be identified across a few different research areas, ranging from parallel algorithms to Pattern Matching and Automatic Speech Recognition (ASR).

Whenever problems can be decomposed in such a way that multiple CE can operate on a problem without having to communicate intensively with each other, such solutions can and have been explored. Just to mention a few areas, Parallel algorithms spread computations across multiple processors; In biologically motivated agent-based approaches, 'colonies' of agents have been used to obtain better and faster solutions in optimization problems [Sachdev, 1998; Talukdar et al., 1997; Maniezzo and Colorni, 1999]; In Pattern Matching, ensembles of classifiers have been used to obtain classifications that are more robust than the most robust individual classifier [Kittler et al., 1998]; in Sensor Data Fusion, the combination of information originated at different sensors results in better recognition than that provided by the best sensor [Rao, 2001]; In Automatic Speech Recognition (ASR), the combination of results of multiple recognizers has been shown to be better than that of the best recognizer [Fiscus, 1997].

Even though work in these fields address the issue of using multiple CE, their focus is localized and techniques are understood either as being tied to a style of programming (parallel computation), or to algorithmic design (biological agents) or to classification problems (pattern matching, ASR), or targeting specific problems (sensor fusion). In particular, approaches that deal with error reduction are usually not concerned with efficiency and vice-versa.

Different goals result in different CE structures, and in different patterns of cooperation, that preclude their use as-is to tackle both objectives at the same time.

We are therefore interested in defining an architectural style, that we call *Distributed Ensembles*, that is based on a unified understanding of the possible uses of multiple CE, and the combination of their partial results. We are particularly interested in exploring 1) performance and scalability through multiprocessing of CE; 2) faster convergence by sharing partial results among CE; 3) error reduction by combination of hypotheses generated by multiple CE.

1.2 Recognition of Human Communication

Human communication is characterized by a wide range of modalities that convey redundant and complementary information. In spoken languages, speech and lip shapes provide information that is redundant, while facial expressions, gestures and body motions complement the main (spoken) modality.

From the point of view of applicability, the technique we want to explore - Distributed Ensembles - multimodal recognition of human communication offers an ideal opportunity. The problem formulation calls for the use of multiple recognizers, that handle input related to different modalities. As we will see, within each recognizer, opportunities abound to apply the proposed approach, particularly, but not restricted to, motion-based modalities that are handled through machine vision techniques.

1.3 American Sign Language

Not much is understood at this point about the semantic of modes other than speech (and closely related ones, such as lip movements). This poses a barrier to the development of full-fledged systems able to tap into the wide range of information that is conveyed by unconstrained gestures and body related expressions in general.

Sign Languages, such as American Sign Language (ASL) also rely on multiple modalities, albeit different from spoken language ones. Sign Languages are mainly motion-based, being exercised through gestures, facial and body expressions. Unlike spoken languages, though, the semantic associated with these modes is precisely specified and linked to standard surface forms (both for gestures and expressions), constrained by lexical and grammatical rules. Sign Languages are not just a code, but living languages that have a community of users for whom they are the primary, native languages. As such, these languages offer an ideal test bed for the development of multimodal techniques that explore motion-based human communicative behavior that incorporates gestures, facial, and body expressions [Vogler and Metaxas, 2001; Bauer, 2001], being amenable to well researched natural language processing techniques.

1.4 Organization of this paper

In the rest of this paper we present background information related to the recognition of human communication, and describe the basic structure of recognizers, issues in multimodal systems and some aspects of the target language, ASL (Section 2). Section 3 discusses approaches that employ multiple CE, their similarities and differences, which serve as inspiration for our work. The proposed style - Distributed Ensembles - is described in Section 4, where we lay out the proposed work.

2 Background on Human Communication Recognition

Human communication is characterized by a wide range of modalities that convey redundant and complementary information. In spoken languages, speech and lip shapes provide information that is redundant, while facial expressions, gestures and body motions complement the main (spoken) modality.

This problem is directly related to natural language processing. Techniques developed for automatic speech recognition are thus directly applicable. Particularly adequate are approaches that employ combinations of multiple hypotheses, that so far have not been explored in this context.

Recognition phases, even for individual recognizers, offer plenty of opportunity for utilizing the architectural style we want to explore, particularly if vision-based methods are employed.

2.1 The motion recognition process

Figure 1 presents a breakout of the phases usually found in systems that do automatic recognition of motion-based (as opposed to spoken) human communicative expression. In these systems, phases may be combined differently, but the functionality offered typically falls within the following blocks:

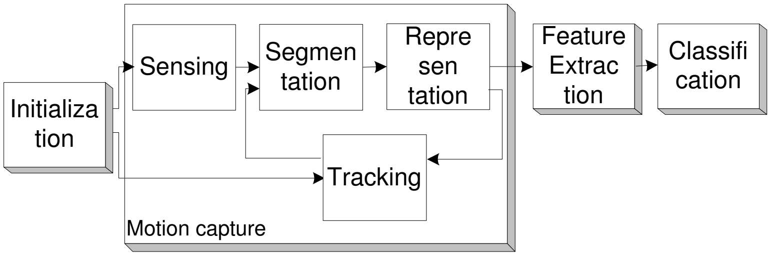

Fig. 1. Generic functional blocks in motion recognition systems. Some systems implement a few of the blocks trivially, or group functionality into different architectural blocks.

- *Initialization* groups activities that need to be performed one time in preparation for the recognition. These activities might include device calibration, or manual initialization of vision-based trackers, and so on.
- *Motion capture* abstracts from a sensed scene only those aspects that are known to carry the bulk of the communicative information, intuitively, those related to the motions of the arms, hands, face and body. Most systems concentrate on a subset of motions, more commonly of the hands, and sometimes fingers, ignoring other important aspects for the sake of simplicity.

  Scenes are sensed at regular intervals dictated by a *sampling rate* and these instantaneous snapshots are then processed to extract information of interest.

  In vision-based systems, snapshots usually correspond to frames grabbed from one or more cameras. Regions of interest are located on these frames and *segmented*. These regions usually correspond to one or two hands and sometimes face. These regions may suffer processing (e.g. dilation, erosion) and are then represented in a more compact way, e.g., as points, edges, splines, contours, blobs or sometimes as kinematics-based models of the body parts (usually hands). Sometimes the representation is based directly on image regions. In this case, representations are built through some dimensionality reduction technique such as e.g. Principal Component Analysis (PCA). Representations might then be fed to a *tracker*, that employs e.g. a Kalman filter, or a condensation filter, for instance, to predict where images of interest might be in subsequence frames, to make the next round of extractions more efficient and robust. Sometimes the tracker applies a much simpler strategy, such as detecting motion by comparing information across frame pairs.

- *Feature extraction* operates (sometimes trivial) transformations on the representations produced by a previous motion capture phase, to further eliminate sources of irrelevant variation.

  Commonly, raw coordinates are adjusted to compensate variations in a person's displacement with respect to the sensing devices, both with respect to distance and displacements in the plane perpendicular to the sensing devices. These adjustments compensate possible variations in persons positions such as their distance to the sensing device, or lateral displacements.

  More sophisticated transformations may be applied, mapping raw data into quantized information, e.g., phoneme-like symbols.
- *Classification* is ultimately responsible for mapping streams of features into streams of decoded symbols, represented according to some lexicon.

  Classifiers need to account for intrinsic variations in the way movements are performed, both in space and in time. Even for a single person, similar common motions are performed with slight variations with respect to trajectory and timing of the movements.

  A few different techniques have been successfully employed in this task, most notably Hidden Markov Models (HMMs), and also Neural Networks (NN) adapted to handle time signals. Other techniques include e.g. instance based learning and clustering techniques.

We next introduce some of the concepts related to Sign Languages in general and their automatic processing in particular.

2.2 Automatic Sign Language recognition

Contrary to a common belief, many different Sign Languages exist, as many in fact as different national languages. In the United States, this language is called American Sign Language, or ASL.

Sign Languages are based on modulations of light, instead of audio signals used in their spoken counterparts. These languages have their own grammar, that match the three-dimensional nature of the gestures that carry the bulk of the linguistic information, and make extensive use of space, for instance to represent pronouns. Grammar and vocabulary, even though influenced by surrounding 'mainstream' spoken languages, are different from these languages and follow their own evolutive path.

Gestures can be highly inflected to represent differences in verbs and to attach adjective-like qualities. Gestures using one (dominant) hand carry the bulk of the information through movements in space and flexion of fingers, that assume positions that correspond to standardized hand shapes and orientations. The other hand plays a secondary but important role, being used in many two-hand signs. Facial expressions and body movements carry far more semantic information than in spoken languages. Certain conventional expressions complement the meaning of hand gestures, and it is in general not possible to determine the meaning of a signed utterance without considering facial expression and sometimes body posture and eye gaze, which convey important aspects of a signed discourse and are not simply coadjuvants that can be ignored.

Automatic recognition systems have been the subject of research in many different parts of the world. Given the challenges that are imposed by the media itself, which has a much larger bandwidth than audio, and the necessary multimodality, existing systems are usually restricted to hand gesture recognition. Even then systems are usually scaled down typically recognizing very small vocabularies that seem to be chosen to fit the limitations of the technology.

Many different approaches are employed to capture and represent information, and to classify it. Systems can be broadly classified into 1) vision-based and 2) instrumentation based. While the former employs cameras to sense the motion, the latter is based on capturing information generated by sensors placed on the body of the signers, usually one or more three-dimensional trackers and many times instrumented gloves that monitor the flexion of fingers.

A good number of systems borrows techniques used in spoken language recognition and employ Hidden Markov Models as the classification mechanism, e.g. [Vogler and Metaxas, 2001; Ma et al., 2000b; Fang et al., 2001; Wang et al., 2001; Bauer et al., 2000; Liang and Ouhyoung, 1998; Starner and Pentland, 1995]. Many other types of classifiers are employed, for instance based on variations of neural networks, (e.g. [Sujan and Meggiolaro, 2000; Yang and Ahuja, 1999; Shin et al., 1999; Abdallah, 1998; Huang and Huang, 1998; Vamplew, 1996; Erenshteyn et al., 1996]); instance based learning [Kadous, 2002; Cui and Weng, 2000], and other techniques, e.g. adaptive fuzzy expert system [Holden and Owens, 2001], template matching [Sutherland, 1996].

Very few systems explore the multimodal nature of the language, exceptions being e.g. [Ma et al., 2000a; Xu et al., 2000; Lu et al., 1997]. The idea of using multiple hypotheses has not been explored in this field, neither has the idea of building systems on top of a distributed infrastructure.

The result of our extensive bibliographic survey of this area can be found at [`http://www.barthelmess.net/Survey_Pages/Sign/Sign_language.html`].

2.3 Multimodal Systems

Communicative modalities can be basically classified into 1) *coupled* and 2) *complementary*. Coupled modes convey roughly the same information simultaneously, e.g. speech and lip movements in spoken languages. Complementary modes operate on related but distinct information. One example in spoken languages would be speech and gestures.

Most commonly, multimodal systems combine speech recognition and lipreading (e.g. in Petajan [1984]; Bregler et al. [1993]), or speech and pen (e.g. Oviatt and Cohen [2000]), but other combinations are also explored for instance: the integration of speech and gestures (e.g. Sowa et al. [1999]); of speech, eye-gaze and hand-gestures (e.g. Koons et al. [1993]); face and gesture (e.g. Clergue et al. [1995]). The context of these solutions is almost invariably that of user interface augmentation by providing additional (multimodal) human-computer channels.

Two main architectural styles are employed in multimodal systems (we draw our description from Oviatt and Cohen [2000]): early fusion systems, that integrate signals at the feature level, and late fusion systems, that integrate information at a semantic level. Representative of the early fusion approach are the systems that employ Hidden Markov Models, in which the model is trained on the two modalities (e.g. lipreading and speech) simultaneously. The recognition process in one mode therefore influences the course of recognition in the other. This kind of design is restricted to coupled mode processing. Late fusion architectures, on the other hand, employ individual recognizer for each modality, whose output is then combined based on time and semantic constraints, thus being adequate for complementary modes. Late fusion also allows in principle the integration of additional modalities to a system in an easier way than is generally possible when early fusion is used.

Some generic modality integration frameworks are proposed, for instance, by Nigay and Coutaz [1993] and Johnston et al. [1997]. This work focus on the integration of individual modalities, and does not contemplate the use of multiple recognizers for similar modalities and the combination of their hypotheses. Also not considered is the potential for performance enhancements through finer grained distribution.

2.4 Multi-hypotheses natural language processing

Multiple hypotheses in the context of Automatic Speech Recognition come in two flavors. The first type of combination is applied to the best hypotheses of multiple independent classifiers, in a post-processing phase that aligns the output using a dynamic programming technique. This technique was first proposed by Fiscus [1997], that gave it the name ROVER, which stands for Recognizer Output Voting Error Reduction. Fiscus showed that a significant error reduction can be obtained when the results of multiple classifiers are combined. Improvements of this technique were proposed e.g. by [Schwenk and Gauvain, 2000b,a].

Another line of research explores hypotheses spaces generated within a single classifier. Conventionally, classifiers based on the dominant HMM technique choose results based on the maximum a posteriori (MAP) probability of an utterance. MAP reflects the maximization of a sentence level probability given acoustic and language models. Some researchers observed that the standard benchmark for language recognition tasks, based on word error rate (WER) is not always minimized by the MAP criterium.

A first attempt to use WER directly as the guiding metric was made in the context of N-best lists [Stolcke et al., 1997]. N-best list approaches explore a larger hypotheses space that is represented by the first few hundreds or thousands of utterance hypotheses, ranked according to MAP. The technique consists of selecting words according to their posterior probabilities in this larger space, so as to minimize the WER.

Extensions of this technique were then developed, that take into account even larger spaces, that correspond to lattices of hypotheses that can be generated by HMM classifiers. The larger hypotheses space results in better solutions, that further minimize the WER with respect for instance to a N-best list approach. Hypotheses lattices are usually very large, and their direct use is therefore usually unmanageable. Techniques proposed by e.g. by Mangu et al. [2000]; Evermann and Woodland [2000]; Wessel et al. [2001]; Goel and Byrne [2000], address this problem by proposing heuristics that approximate the solution, at a lower computational cost.

Other multiple hypotheses approaches to natural language processing have been explored e.g. in the context of parsing [Henderson and Brill, 2000, 1999], that use bagging to enhance results.

3 Ensemble-based solutions

Whenever problems can be decomposed in such a way that multiple CE can operate on the same problem without having to communicate intensively, ensemble-based solutions can be made to work.

Even though solutions share the basic characteristic of employing multiple CE (and the hard nature of the problems that are tackled), the specific nature of the entities, of the communication that is expected or allowed to occur and even the overall goals for using such approaches vary from field to field.

We are interested in combining some of these views, particularly the ones related to error reduction and related to efficient and scalable computation, as well as the role that sharing partial solutions can have on solution convergence.

In this section we aim at presenting these different perspectives and eventually wish to arrive at a unified view that can lead to a useful architectural style that we then explore in further sections of this paper.

Three facets of use of multiple computational elements seem particularly appropriate for our purposes:

1. *Performance and scalability* can be obtained through distributing computation across multiple processors;
2. *Convergence of solutions* can be obtained through sharing of partial solutions, so that individual computations can be redirected into more fruitful directions;
3. *Error reduction* can be obtained by combining different hypotheses produced by multiple computational entities.

We next discuss each of these facets in further detail along with the fields from which they emerge.

3.1 Performance and scalability

Performance and scalability are aspects targeted by parallel high performance computing. The objective is to achieve speedup through simultaneous execution across multiple processors employing special purpose massively parallel machines, multiprocessor machines, clusters of workstations and even wide-area solutions forming computational grids.

Many times, entities that are executed in parallel are symmetric, i.e., except perhaps for instances that have coordination duties, the code that is executed in parallel is identical. This is a requirement in SIMD (Single Instruction Multiple Data) approaches, but is also common in MIMD (Multiple Instructions Multiple Data) capable systems as well, that many times employ computational entities that result from parallelization of a single source code through automatic compiler parallelization (e.g. HPF [`http://www.crpc.rice.edu/HPFF/`]), program annotations (e.g. in OpenMP [`http://www.openmp.org/`]) or calls to a communication API (e.g. MPI [`http://www.mpi-forum.org/`]). Many of these solutions are linked to developments in programming languages, thus operating at a (sub-architectural) level that is close to the machines, and exploit detailed domain specific opportunities for parallelization.

Important aspects highlighted by solutions in this field are the importance of finer grained parallelism and efficient communication infrastructure.

3.2 Collaborating multi-algorithms

Exchanging partial solutions can be a powerful tool to achieve faster convergence, by reorienting individual ongoing computations according to more promising results produced by other entities.

Collaborations of this kind are employed for instance by some biologically motivated solutions, e.g.[Talukdar et al., 1997; Sachdev, 1998; Maniezzo and Colorni, 1999], that usually address hard optimization problems, such as Traveling Sales Person (TSP) or Job Shop Scheduling, for example.

Even though the emphasis is on a relative independence of individual entities, and the simplicity of the cooperation ’behaviors’ they embed, entities benefit from exchanges of partial solutions that are exposed in mid-run, rather than at the end of computation.

Overall cooperation is expected to emerge from simple local rules that regulate interchanges between entities; these interchanges are many times data-centric, in the sense that entities are most of the time unaware of each other except for ’traces’ that are left by other entities as they go about doing their jobs. These ’traces’ correspond to partial results that are exposed and can serve the purpose of reorienting individual actions into more fruitful directions.

Entities might be *symmetric*, i.e., they might have the same structure, and explore different regions of a data space, or they might be *asymmetric*, in which case each one might implement a different algorithm for the same problem, or might employ different parameters of the model that drives the algorithm (e.g. different seeds in a Simulated Annealing algorithm [Chen and Taylor, 2002]).

Results show that faster convergence and better results can be obtained by combining multiple entities than it would be possible to obtain from any of the individual entities that are part of the ensemble. It is interesting to observe that even slow and imperfect algorithms can contribute to an overall solution, that can be made faster and/or more precise by their inclusion in an ensemble.

Techniques are closely tied to optimization problems. Exchange of solutions is in many cases facilitated by the existence of objective measurements of goodness, that make it possible to quickly and inexpensively identify which solutions are the most promising at any moment.

The use of multiple processors, even if perhaps natural, is not emphasized, due perhaps to the added coordination complexity that would be required, which might clash with the desire to concentrate on algorithmic solutions, rather than on solutions where coordination aspects play a stronger role.

3.3 Multi-hypotheses approaches

Multi-hypotheses approaches are used in a few areas, such as Pattern Matching (using ensembles of classifiers [Kittler et al., 1998; Kuncheva, 2002], Sensor Data Fusion [Rao, 2001; Crowley and Berard, 1997; Azoz et al., 1998] and in Automatic Speech Recognition (in combinations of recognizers output [Fiscus, 1997] or in the context of n-best lists [Stolcke et al., 1997] or hypotheses lattice processing [Mangu et al., 2000].

Use here is connected to classification tasks, and the hypotheses result from the use of multiple classifiers, or from multiple hypotheses originated by a single classifier (e.g. a Hidden Markov Model - HMM), as these classifiers explores multiple alternative interpretations of evidence.

Two categories of CE (classifiers in this case) can be identified: 1) CE that accept the same representations/features as input, and differ in the parameters of the models that guide classification, e.g. if bagging is used [Bauer and Kohavi, 1999], and 2) CE that use different input representations as features (and therefore other models as well). One example is the use of data generated by multiple sensors that target a common scene, but employ different technology (e.g. infrared, radar, cameras).

The distinguishing trait here is that the same underlying phenomenon that one wants to classify (recognize) elicits multiple possible interpretations. These different hypotheses are combined, usually resulting in solutions that are better than that of the best individual classifier/sensor.

A good amount of work of the Pattern Recognition community has been applied to combining classifiers to achieve better recognition. The maturity of this sub-area is reflected in work that explores the reasons why certain combinations of classifiers are more successful than others, e.g. Kittler et al. [1998]; Kuncheva [2002]; Bauer and Kohavi [1999].

A requirement for successful combination is that errors of constituent CE are minimally overlapping. In other words, the distribution of errors must be such that a significant number of constituent CE agree on the correct solution most of the time with high probability, and conversely, disagrees on wrong solutions with high probability as well.

Many different types of combination strategies are possible, ranging from majority voting, weighted voting, to complex combinations in which classifiers are organized in multiple stages, either sequentially, or pipelined or hierarchically. The first stages can for example use small sets of cheap features, and latter stages employing more complex algorithms and features [Kittler et al., 1998].

Combination of hypotheses usually takes place in a subsequent, post-processing step that organizes alternatives and reaches final decisions after all entities have finished producing the hypotheses by working in isolation.

Distribution, whenever considered, happens at a coarse granularity, at the level of whole recognizers in speech recognition systems, for example.

3.4 Discussion

Despite the common trait of using multiple CE, that we identify as being the hallmark of what we claim is a family of related approaches, individual fields have developed solutions that target different goals. Objectives are high performance through parallelism, faster convergence of solutions through different algorithms and more reliable solutions through the combination of hypotheses.

These different objectives reflect in different CE design choices, some of which we highlight next and summarize in Table 1:

- Granularity of distribution : depending on the objective, sections of code that correspond to multiple instances can be finer or coarser. In particular, finer grained solutions are employed to obtain high performance; the granularity increases as we move away from this objective into multi-algorithms and multi-hypotheses. Particularly the latter approach tends to use coarse grained CE.
- Breakdown of a problem : sometimes multiple CE operate on distinct regions of a data space, e.g. different parts of a matrix that is being subjected to some transformation. In other occasions, there is no clearcut separation into regions: in optimization problems, for instance, different algorithms may explore regions of a search space by using different methods, and these searches might sometimes overlap at least partially. Multi-hypotheses approaches employ a perhaps even tighter overlap of regions that are explored by each CE.
- Cooperation among CE : faster convergence through the use of ensembles of algorithms is based on the effects of cooperation. Multiple CE reorient their computations according to partial results produced by other CE, thus speeding up their own computations. High performance approaches sometimes cooperate indirectly through sharing of data structures, e.g., matrix operations performed by a CE might involve the use of neighboring data values produced by other CE. Multi-hypotheses approaches tend not to employ direct cooperation among CE, but rather to combine results in a post-processing phase.
- Synchronism: parallel computations are usually associated with tight synchronism among CE, aiming at lock-step execution whenever possible. Multi-algorithms are characterized by an intentional low level of synchronism among CE. Ideally, execution is asynchronous except for potential side-effects of sharing of partial results. Multi-hypotheses approaches assume asynchronous execution of CE, that are for all purposes unaware of the existence of other CE.
- Coordination: parallel computations are coordinated most of the time from within code that embeds both parallel and serial code. Multi-algorithmic solutions usually require some external coordination mechanism that handles the distribution of partial solutions, for instance employing a blackboard model. Multi-hypotheses also require coordination support that collects individual results and presents them to a subsequent phase for combination.

Despite the differences, there is no intrinsic reason why these different aspects cannot be combined in a unified solution. In the next section we explore some of the implications of these combinations in the context of the work we propose.

	High performance	Multi-algorithmic	Multi-hypotheses
Granularity	Fine	Medium	Coarse
Breakdown	Distinct regions	Overlap	Overlap
Cooperation	Shared data	Explicit	Post-processing
Synchronism	Lock-step	Indirect	Asynchronous
Coordination	Embedded	Mixed	External

Table 1. Differences between ensemble-based approaches along a few dimensions.

4 Proposed work: Distributed Ensembles

Once the commonalities of the approaches we just described are identified and one understands them as facets of a unified technique, it is possible to rethink implicit assumptions and come up with a technique that combines the strengths of these different facets. *Distributed Ensembles* is the name we give the style that is based on this combination.

The key observation for our purpose is that the goals of error reduction and higher performance are usually not associated, representing goals of techniques at the extremes of the spectrum (multi-hypotheses classification and parallel computation). The common traits shared by all the techniques in different fields is that multiple computational elements are employed to solve the same problems. On the one hand, the fact that it is possible to identify these multiple elements is conducive to their distribution for enhanced performance; furthermore, one can take advantage of the different combination strategies that result in higher quality solutions.

Our goal is therefore to explore these ideas as a uniform architectural style that is characterized by:

- Enhanced performance through fine granularity of distribution of multiple computational elements. Finer granularity promotes distribution among a larger number of processors, thus resulting in potential performance enhancement.
- Cooperative behavior by sharing of partial results among CE, to promote informed pruning.
- Higher quality of results through the combination of partial or alternative results produced by individual computational elements.

The class of problems that is bound to benefit from such an architectural style is relatively broad, and consists of those problems where the gains in performance and quality of results are high enough to compensate for communication overhead among CE.

We want to explore these issues in the context of an area that seems particularly promising in terms of application of this new architectural style - the recognition of human communication.

4.1 Distributed ensembles for motion-based human communication recognition

Some of the opportunities for applying a Distributed Ensembles approach to motion-based human communication recognition are:

- Processing of multiple stereo images - stero images are built from multiple frames, captured by two or more cameras. The processing required for building the stereoscopic view can be assigned to multiple CE, distributed across processors for speedũp.

- Tracking of regions of interest - tracking can be made more robust if multiple cues are combined (e.g. face tracking using blink detection, normalized color histogram matching, and cross correlation (SSD and NCC) [Crowley and Berard, 1997]). These cues can be profitably handled by multiple CE that can be distributed for enhanced performance.
- Segmentation of regions of interest - segmenting regions that correspond to hands, face can be cast as a classification problem, where pixels are classified either as belonging to one of the regions of interest or to a background. Classification problems are particularly prone to benefit from multi-hypotheses approaches and therefore are good candidates for our technique as well.
- Feature extraction - features are sometimes clustered or quantized, in a process that again corresponds to classification, from raw features into phoneme-like ones, for example.
- Lexical classification - classification of features into lexical items is the foremost example of a phase where multiple-hypotheses approaches can be (and have been) used.

Distributed Ensembles have thus a potential for being used to reduce error and enhance performance in all these phases and in others where similarly work can be executed by ensembles of CE.

4.2 Distributed word hypotheses processing

Given the wide scope of the overall recognition process, we cannot hope to be able to apply the architectural style to all phases of a recognition system at this time, even though this is our longer term goal. We therefore choose to concentrate on exploring Distributed Ensembles in a specific phase - the combination of multiple hypotheses generated by classifiers (Figure 2).

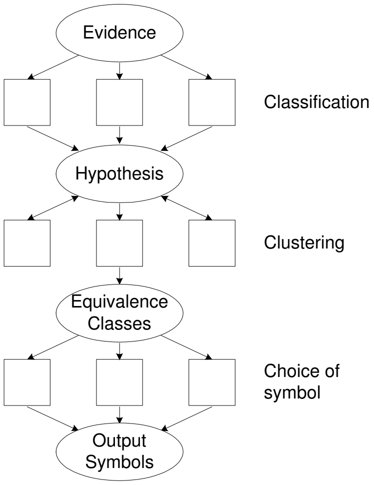

Fig. 2. Multi-hypotheses classification. Boxes are computational entities and ovals are communication media, that can be thought for the sake of discussion as blackboards.

Combination of hypotheses can be potentially employed in multiple phases of the overall process, for instance in tracking, segmentation, feature extraction and lexical classification. Here we consider it in the context of lexical classification. This is a well-researched subproblem that has attracted considerable recent attention in the context of ASR (e.g. Mangu et al. [2000]; Evermann and Woodland [2000]; Stolcke et al. [1997]; Wessel et al. [2001]; Goel and Byrne [2000]).

Consider the non-parallel processing of word lattices as proposed by Mangu et al. [2000]. Suppose a classifier has produced a lattice containing alternative hypotheses generated during recognition of an utterance (Figure 3).

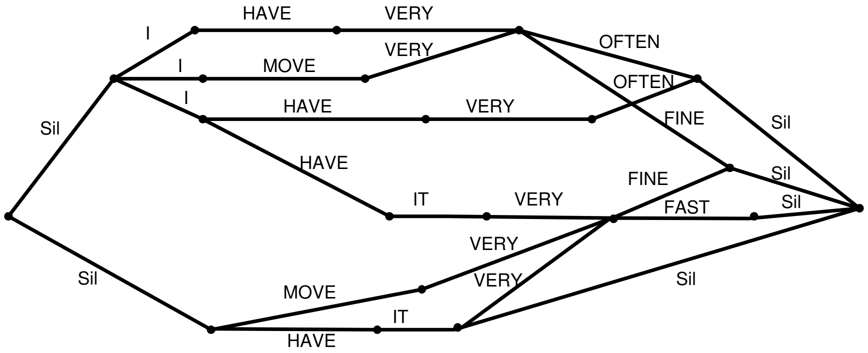

Fig. 3. Recognizer's word hypotheses graph. Each path defines a potential interpretation of the evidence. (from Mangu et al. [2000]).

Selecting symbols directly from the lattice is usually unmanageable, due to the large size of the hypotheses spaces. One approach consists of applying heuristics to convert a lattice into a more compact representation, a *confusion network* [Mangu et al., 2000]. Figure 4 presents a confusion network for the lattice presented in Figure 3.

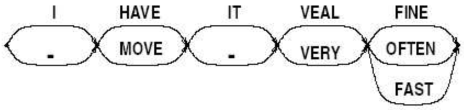

Fig. 4. Confusion networks are compact representations of hypotheses lattices (from Mangu et al. [2000]).

Confusion networks represent alignments that consist of equivalence relations over a lattice word hypotheses. The total ordering of the equivalence classes that characterize a confusion network is consistent with that of the original lattice. Each equivalence class contains alternative words and the final recognition result can be obtained by choosing one word from each such class, in the order they appear in the confusion network.

Given multiple classifiers, different classifiers potentially generate different hypotheses from each other. Depending on the error distribution of these classifiers, the combination of these multiple alternative hypotheses has a potential for reduction of error by reinforcing correct hypotheses and disqualifying incorrect ones (that would then be pruned out).

The heuristic for determining membership in equivalence classes is based on: 1) the original lattice's partial order, 2) word similarity, 3) time overlap, weighted by the words posterior probabilities.

We want to explore a parallel, fine-grained implementation of this approach, by having the classifiers output hypotheses as soon as they are produced rather than at the end of utterances.

We want to explore the effects of cooperation by providing each classifier with access to partial solutions produced by others to prune their own search. The objective is to produce a unified compact representation of hypotheses that is built on-line, as the multiple classifiers sweep a common input in a time-synchronous fashion.

There is indication that an early combination has a potential for both reduction of the effort employed, by discarding hypotheses that are known to have low probability and enhance the precision of the solution, by avoiding pruning of hypotheses on which classifiers agree (see for instance initial work reported by Zheng and Yan [2002]).

4.3 Enabling the style through an infrastructure

Architectural design takes into consideration the overall structure of a system, determining how basic entities cooperate and coordinate. By abstracting the coordination aspect of a design, systems' components can be made simpler. Each of the basic entities of a system can be defined purely in terms of the computations that are required, without concerns for how coordination is achieved.

The coordination aspect needs of course to be provided by other parts of a system, once it is factored out of basic entities. Not uncommonly, given the generic nature of coordination, this aspect is provided by an *architectural infrastructure*.

Architectural infrastructures provide the functionality that allows other parts of a system to work together. These infrastructures usually take the form of a software layer that implements the required functionality. The services provided by this layer are usually exposed through Application Program Interfaces (API), that expose an abstracted view that is used by system entities to access these services.

Architectural infrastructures are enablers of specific architectural styles. Without this infrastructure, the cost of embedding coordination aspects into each element of a system can become prohibitive.

A very large number of architectural infrastructures exist. We focus here on a few that specifically target the processing of human communication.

Existing infrastructures, such as the DARPA Communicator, based on MIT's Galaxy Architecture [Mitre Corporation, 2002], for instance, enable a style where coarse granularity entities can be distributed. Entities in this case are for example parsers, speech recognizers, dialog managers and similar ones. In DARPA Communicator, these entities are called *servers* and communicate through a centralized *hub*. The hub receives messages from the servers and activates services of these servers based on a rule-like script. The hub concentrates all communication and thus presents a potential bottleneck, particularly if a style requiring more communication is to be employed. Connections between servers and hub are based on TCP/IP sockets and might not be adequate if larger data structures need to be shared. Even though servers can be distributed, no advantage is taken of facilities present in multiprocessor machines, such as hardware supported shared memory. The style that DARPA Communicator is meant to support in therefore one in which coarser granularity entities execute mostly in isolation, and messages are infrequent (compared to the overall processing time) and preferably small.

The Open Agent Architecture [Martin et al., 1999; Cheyer and Martin, 2001] promotes a similar style to DARPA Communicator's, that concentrates coordination duties into a single broker, called the facilitator.

The Distributed Application's Communication System (DACS) [Fink et al., 1996] is an architectural infrastructure that targets multimodal systems that include e.g. speech and vision-based processing. The infrastructure is based on daemons that reside on each machine and handle the communication on behalf of the processes that reside on each machine. A central naming service allows for dynamic reconfiguration, e.g. the addition of viewers to monitor system operations. Heterogenous machines can be used and the infrastructure performs marshaling and un-marshaling of data based on common reified data structure representation (NDR, or Network Data Representation). This common structure is explored, for instance, by generic debugging tools that can examine messages as they flow through the daemons.

What these infrastructures have in common is the decoupling of coordination from processing, that allows for easy system reconfiguration (either static or dynamic, depending on the infrastructure).

From our perspective, these infrastructures lack some of the functionality that we envision is necessary for supporting a Distributed Ensembles style:

- The finer granularity proposed by the style implies that more entities will be distributed over a potentially larger number of processors, which might add to the complexity of launching and monitoring.
- A larger number of entities also implies that a possibly larger volume of communication will take place over shorter periods of time. As a consequence, communication needs to be optimized or it might become a bottleneck.
- Cooperative behavior might require sharing of potentially large data structures, that might not fit well into a strictly message-based paradigm.

Given the above mentioned requirements, we want to explore high performance techniques to enhance communications and cooperation among larger number of finer grained computational entities, taking advantage of clusters of potentially multiprocessed workstations.

High performance technology takes advantage of hardware and system level strategies that bypass lengthier network-based communications, resulting in potential efficiency that we expect will result in adequate support for the style we want to pursue.

4.4 Validation

We plan to validate our claims along two different dimensions:

- Usefulness of the style: we plan to develop an ASL recognition system, and employ the Distributed Ensembles style in a particular phase, the processing of hypotheses, mainly in the lexical classification. This processing, when performed, is done as a post-processing step over lattices produced by independent recognizers. A final post-processing step combines 1-best results of these independent recognizers. As described in Section 4.2, we plan to explore a finer grained, integrated processing that can take advantage of cross-influence between recognizers.
- Performance improvements: we plan to compare the performance of recognition using the proposed approach and infrastructure with a baseline that employs the conventional post-processing approach.

Bibliography

Abdallah, M.: 1998, ‘A neuro-hierarchical multilayer network in the translation of the American sign language’. In: *Proceedings of the IEEE Southeastcon:Region 3*. pp. 224–227.

Azoz, Y., L. Devi, and R. Sharma: 1998, ‘Reliable Tracking of Human Arm Dynamics by Multiple Cue Integration and Constraint Fusion’. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Bauer, B.: 2001, 'Towards an Automatic Sign Language Recognition System Using Subunits'. In Wachsmuth and Sowa [2001], Springer-Verlag. Draft.

Bauer, B., H. Hienz, and K.-F. Kraiss: 2000, 'Video-based Continuous Sign Language Recognition Using Statistical Methods'. In: *Proceedings of the International Conference on Pattern Recognition*, Vol. II. pp. 463–466.

Bauer, E. and R. Kohavi: 1999, 'An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants'. *Machine Learning* **36**(1-2), 105–139.

Bregler, C., H. Hild, S. Manke, and A. Waibel: 1993, 'Improving Connected Letter Recognition by Lipreading'. In: *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*.

Chen, J. and V. E. Taylor: 2002, 'Mesh Partitioning for Efficient Use of Distributed Systems'. *IEEE Transactions on Parallel and Distributed Systems* **13**(1).

Cheyer, A. and D. Martin: 2001, 'The Open Agent Architecture'. *Journal of Autonomous Agents and Multi-Agent Systems* **4**(1/2).

Clergue, E., M. Goldberg, N. Madrane, and B. Merialdo: 1995, 'Automatic face and gestural recognition for video indexing'. In: *International Workshop on Automatic Face- and Gesture-Recognition*.

Crowley, J. L. and F. Berard: 1997, 'Multi-Modal Tracking of Faces for Video Communications'. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Cui, Y. and J. Weng: 2000, 'Appearance-Based Hand Sign Recognition from Intensity Image Sequences'. *Computer and Vision Image Understanding* **78**, 157–176.

Erenshteyn, R., P. Laskov, R. Foulds, L. Messing, and G. Stern: 1996, 'Recognition Approach to Gesture Language Understanding'. In: *Proceedings of the International Conference on Pattern Recognition*. Vienna.

Evermann, G. and P. Woodland: 2000, 'Posterior Probability Decoding, Confidence Estimation and System Combination'. In: *Proceedings Speech Transcription Workshop*.

Fang, G., W. Gao, X. Chen, C. Wang, and J. Ma: 2001, 'Signer-independent continuous sign language recognition based on SRN/HMM'. In Wachsmuth and Sowa [2001], Springer-Verlag.

Fink, G., N. Jungclaus, F. Kummert, H. Ritter, and G. Sagerer: 1996, 'A Distributed System for Integrated Speech and Image Understanding'. In: *International Symposium on Artificial Intelligence*.

Fiscus, J.: 1997, 'A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (rover)'. In: *Proceedings IEEE Workshop on Automatic Speech Recognition and understanding*. Santa Barbara, USA.

Goel, V. and W. Byrne: 2000, 'Minimum Bayes-risk automatic speech recognition'. *Computer Speech and Language* **14**(2), 115–135.

Henderson, J. C. and E. Brill: 1999, 'Exploiting Diversity in Natural Language Processing: Combining Parsers'. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. College Park, Maryland.

Henderson, J. C. and E. Brill: 2000, 'Bagging and Boosting a Treebank Parser'. In: *NAACL*.

Holden, E. J. and R. Owens: 2001, 'Visual sign language recognition'. In: R. Klette, T. Huang, and G. Gimen'garb (eds.): *10th International Workshop on Theoretical Foundations of Computer Vision*, Vol. 2032 of *Lecture Notes in Computer Science*. Springer. Also in Proceedings of DICTA'99 (Digital Image Computing: Techniques & Applications), pp. 275-279.

Huang, C. and W. Huang: 1998, 'Sign Language Recognition Using Model-Based Tracking and a 3D Hopfield Neural-Network'. *Machine Vision and Applications* **10**(5/6), 292–307.

Johnston, M., P. R. Cohen, D. McGee, S. L. Oviatt, J. A. Pittman, and I. Smith: 1997, 'Unification-based Multimodal Integration'. In: *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*.

Kadous, M. W.: 2002, ‘Temporal Classification: Extending the Classifiation Paradigm to Multivariate Time Series’. Ph.D. thesis, The University of New South Wales, School of Computer Science and Engineering.

Kittler, J., M. Hatef, R. P. Duin, and J. Matas: 1998, ‘On combining classifiers’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 226–239.

Koons, D., C. Sparrell, and K. Thorisson: 1993, ‘Integrating simultaneous input from speech, gaze, and hand gestures’. In: *Intelligent Multimedia Interfaces*. MIT Press, Chapt. 11.

Kuncheva, L. I.: 2002, ‘A Theoretical Study on Six Classifier Fusion Strategies’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(2), 281–286.

Liang, R. and M. Ouhyoung: 1998, ‘A real-time continuous gesture recognition system for sign language’. In: *IEEE International Conference on Automatic Face and Gesture Recognition*. pp. 558–567.

Lu, S., K. Imagawa, and S. Igi: 1997, ‘An Gazing-line Generation System for Improving Sign Language Conversation’. In: *Proc. of 7th International Conference on Human-Computer Interaction*, Vol. 2. California, USA, pp. 283–286.

Ma, J., W. Gao, and R. Wang: 2000a, ‘A parallel multistream model for integration of sign language recognition and lip motion’. In: T. Tan, Y. Shi, and W. Gao (eds.): *Proceedings of the Third International Conference on Advances in Multimodal Interfaces*, Vol. 1948 of *Lecture Notes in Computer Science*. pp. 582–589.

Ma, J., W. Gao, J. Wu, and C. Wang: 2000b, ‘A Continuous Chinese Sign Language Recognition System’. In: *IEEE International Conference on Automatic Face and Gesture Recognition*.

Mangu, L., E. Brill, and A. Stolcke: 2000, ‘Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks’. *Computer, Speech and Language* **14**(4), 373–400.

Maniezzo, V. and A. Colorni: 1999, ‘The Ant System Applied to the Quadratic Assignment Problem’. *IEEE Transactions on Knowledge and Data Engineering* **11**(5).

Martin, D., A. Cheyer, and D. Moran: 1999, ‘The Open Agent Architecture: a framework for building distributed software systems’. *Applied Artificial Intelligence* **13**(1/2).

Mitre Corporation: 2002, ‘Galaxy Communicator Documentation’. Available on the web at `http://communicator.sourceforge.net/sites/MITRE/distributions/GalaxyCom%municator/docs/manual/index.html`.

Nigay, L. and J. Coutaz: 1993, ‘A Design Space For Multimodal Systems: Concurrent Processing and Data Fusion’. In: *Proceedings of INTERCHI ’93*.

Oviatt, S. and P. Cohen: 2000, ‘Multimodal systems that process what comes naturally’. *Communications of the ACM* **43**(3).

Petajan, E.: 1984, ‘Automatic lipreading to enhance speech recognition’. In: *Proceedings of the IEEE Communication Society Global Telecommunications Conference*.

Rao, N. S.: 2001, ‘On Fusers that Perform Better than Best Sensor’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(8), 984–909.

Sachdev, S.: 1998, ‘Explorations in Asynchronous Teams’. Ph.D. thesis, Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA.

Schwenk, H. and J.-L. Gauvain: 2000a, ‘Combining Multiple Speech Recognizers using Voting and Language Model Information’. In: *IEEE International Conference On Speech and Language Processing*. pp. II:915–918.

Schwenk, H. and J.-L. Gauvain: 2000b, ‘Improved ROVER using Language Model Information’. In: *ISCA ITRW Workshop Automatic Speech Recognition*. pp. 47–52.

Shin, S.-H., S.-W. Kim, and Y. Aoki: 1999, ‘A structural learning of MLP classifiers using PfSGA and its application to Korean sign language recognition’. In: *Proceedings of the IEEE Region 10 Conference TENCON 99*. pp. 190–193.

Sowa, T., M. Frohlich, and M. E. Latoschik: 1999, 'Temporal Symbolic Integration Applied to a Multimodal System Using Gestures and Speech'. In: *Lecture Notes in Computer Science*. Gif-sur-Yvette, France, pp. 291–302, Springer-Verlag.

Starner, T. and A. Pentland: 1995, 'Visual recognition of american sign language using hidden markov models'. In: *International Workshop on Automatic Face and Gesture Recognition (IWAFGR)*. Zurich, Switzerland.

Stolcke, A., Y. Konig, and M. Weintraub: 1997, 'Explicit Word Error Minimization in N-Best List Rescoring'. In: *Proc. Eurospeech '97*. Rhodes, Greece, pp. 163–166.

Sujan, V. A. and M. A. Meggiolaro: 2000, 'Sign language recognition using competitive learning in the HAVNET neural network'. In: N. M. Nasrabadi and A. K. Katsaggelos (eds.): *Applications of Artificial Neural Networks in Imaging V (Electronic Imaging 2000)*, Vol. 3962 of *Proc. SPIE*. San Jose, CA, pp. 2–12.

Sutherland, A.: 1996, 'Real-time video-based recognition of sign language gestures using guided template matching'. In: P. A. Harling and A. D. Edwards (eds.): *Proceedings of Gesture Workshop*. University of York, UK, pp. 31–38, Springer-Verlag.

Talukdar, S., S. Sachdev, and E. Camponagara: 1997, 'A Collaboration Strategy for Autonomous, Highly Specialized Agents'. In: *Proceedings of the SPIE, Symposium on Intelligent Systems & Advanced Manufacturing*.

Vamplew, P.: 1996, 'Recognition of Sign Language Using Neural Networks'. Ph.D. thesis, Department of Computer Science, University of Tasmania.

Vogler, C. and D. Metaxas: 2001, 'A Framework for Recognizing the Simultaneous Aspects of American Sign Language'. *Computer and Vision Image Understanding* **3**, 358–384.

Wachsmuth, I. and T. Sowa (eds.): 2001, 'Proceedings of Gesture Workshop', No. 2298 in Lecture Notes in Computer Science. London, UK: Springer-Verlag.

Wang, C. L., W. Gao, and Z. G. Xuan: 2001, 'A real-time large vocabulary continuous recognition system for Chinese Sign Language'. In: *IEEE Pacific Rim Conference on Multimedia 2001*, Vol. 2195 of *Lecture Notes in Computer Science*. Bejing, China.

Wessel, F., R. Schlter, and H. Ney: 2001, 'Explicit Word Error Minimization Using Posterior Word Hypothesis Probabilities'. In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. Salt Lake City, USA.

Xu, M., B. Raytchev, K. Sakaue, O. Hasegawa, A. Koizumi, M. Takeuchi, and H. Sagawa: 2000, 'A Vision-Based Method for Recognizing Non-manual Information in Japanese Sign Language'. In: *Third International Conference on Multimodal Interfaces*, Vol. 1948 of *Lecture Notes in Computer Science*. Beijing, China, pp. 572–581, Springer.

Yang, M.-H. and N. Ahuja: 1999, 'Recognizing Hand Gesture Using Motion Trajectories'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Ft. Collins, CO, pp. 466–472.

Zheng, C. and Y. Yan: 2002, 'Run Time Information Fusion in Speech Recognition'. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*. Denver, Colorado.