Soft Comput (2008) 12:657–665
DOI 10.1007/s00500-007-0244-1

FOCUS

Conceptual duplication

Soft-clustering and improved stability for adaptive resonance theory neural networks

Louis Massey

Published online: 28 September 2007

Abstract Stability and plasticity in learning systems are both equally essential, but achieving stability and plasticity simultaneously is difficult. Adaptive resonance theory (ART) neural networks are known for their plastic and stable learning of categories, hence providing an answer to the so called stability-plasticity dilemma. However, it has been demonstrated recently that contrary to general belief, ART stability is not possible with infinite streaming data. In this paper, we present an improved stabilization strategy for ART neural networks that does not suffer from this problem and that produces a soft-clustering solution as a positive side effect. Experimental results in a task of text clustering demonstrate that the new stabilization strategy works well, but with a slight loss in clustering quality compared to the traditional approach. For real-life intelligent applications in which infinite streaming data is generated, the stable and soft-clustering solution obtained with our approach more than outweighs the small loss in quality.

Keywords Adaptive resonance theory · Stable learning · Neural networks · Machine learning

1 Introduction

Stability is an essential aspect of learning; without it, an intelligent system becomes subject to catastrophic forgetting. There are two types of stability: the first one is stable attribution of data. This means that if an identical datum is presented several times to a learning system, it will be consistently recognized as belonging to the same category. For instance, a circle should continue to be recognized as a circle in a shape recognition system and a robin as a bird in an animal classification application. The second type of stability is one that ensures that given a finite data set repetitively presented to a learning system, there will not be endless proliferation of categories. For example, in the case of a parts recognition system in a manufacturing environment, the parts previously classified by the system should not continually trigger the formation of new (and most likely useless) part types when the same parts are presented again. A learning system is deemed stable if both types of stability are achieved.

One could describe stability as being about remembering past experiences and avoiding changes. On the other hand, another very important aspect of learning is plasticity, one that defines adaptability to new situations. Plasticity, contrary to stability, is the property of learning systems that allows for continuous learning in the face of novelty. Stability and plasticity are forces that conflict. Consequently, it is rather difficult to achieve both simultaneously in artificial learning systems, although clearly natural learning systems do not seem to suffer from this problem.

It is trivial for an on-line learning system to be stable: it merely has to stop learning on new data, for instance by iteratively decreasing the value of a learning parameter such that as time passes, less and less learning takes place. On the other hand, off-line supervised classification learning systems achieve stability by forfeiting plasticity all together. Indeed, once the classifier function has been acquired, no new learning is allowed. In both cases, one had to give up on plasticity to achieve stability.

Adaptive resonance theory (ART) neural networks (Grossberg 1976; Carpenter and Grossberg 1995) have been

This research was supported in part by the National Defence Academic Research Program (ARP) under grant 743321.

L. Massey (✉)
Department of Mathematics and Computer Science,
Royal Military College of Canada,
Kingston, ON K7K 7B4, Canada
e-mail: massey@rmc.ca

designed over 30 years ago by Stephen Grossberg to address this exact problem of constructing learning systems that are both plastic and stable. ART networks properties of stability and plasticity as well as their ability to process dynamic data efficiently make them attractive candidates for recognizing patterns in large, rapidly changing data sets generated in real-life environments. The applications of ART span many domains, including among others sonar signal recognition (Carpenter and Streilein 1998), parts management at Boeing (Caudell et al. 1991) and text clustering (Massey 2003).

ART networks are on-line, unsupervised learning systems, allowing both continuous learning (plasticity) and guaranteeing a stable internal representation. ART converges to a stable representation after at most $R-1$ presentations of the R data items (Georgiopoulos et al. 1990). However, an important and until now unresolved problem with ART stability was recently identified while investigating its application to a real-world problem (Massey 2005b). In short, the problem is that contrary to general belief, ART stability is not possible with infinite streaming data. In order for ART to be usable in a real-life environment characterized by a continuous data stream and by periodic novelty detection capability, a solution to this problem is imperative.

The work presented here is very different from our previous work with ART (Massey (2002, 2003, 2005a, b) where we tested a standard version of ART under various conditions of text clustering and measured the effectiveness of documents grouping by topics for real-life applications. Here, we present and analyse in detail the ART stabilisation problem we have previously identified briefly in Massey (2005b). Our contribution in this paper is to resolve this problem by presenting and testing a new stabilization strategy called *conceptual duplication*. The Conceptual Duplication principle offers two major advantage over regular ART stabilization : (1) an actually stable representation for infinite streaming data common in real-life applications; and, (2) a soft-clustering solution compatible with realistic classification of text documents.

In Sect. 2 of this paper, we describe ART neural networks, including their standard stabilization process, how this process fails to deliver on its promise of stability and then how conceptual duplication works to resolve the problem. In Sect. 3, we experiment with and discuss the new stabilization strategy in a topics recognition task using a benchmark text corpus.

2 Adaptive resonance theory

2.1 Description of ART networks

Definition 1 (*Data set*) Let $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_R}\}$ be a set of individual datum (data element or object). X is the data set of cardinality R of all such data elements, each data element being a vector in $\{0, 1\}^N$ of the form $\mathbf{x_k} = (x_1, x_2, \ldots, x_N)$.

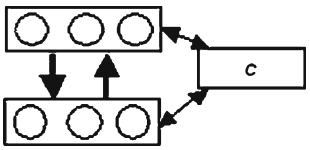

Fig. 1 The ART1 architecture showing the two interconnected layers of neurons and the external control system C

Definition 2 (*Cluster or category*) A cluster (category) is a subset $\alpha \subseteq X$. A clustering solution is the set of clusters a clustering algorithm discovers in a data set X. Hard clustering is a partition of X into mutually exclusive clusters while soft clustering allows a datum to belong to more than one cluster.

In this paper we focus on the binary ART version known as ART1. The general architecture of an ART1 network is summarized in Fig. 1. The network is made of two interconnected layers of neurons and of an external control system (the box labeled C at the right of Fig. 1) that determines the operational mode of the layers. Weights w_{ij} exist on bottom-up connections going from input neuron i to output neuron j. There is one input neuron i for each component of an input vector $\mathbf{x_k}$ of dimension N. Weights t_{ji} are attributed to top-down connections, from output neuron j to input neuron i.

Definition 3 (*Prototype*) Each output neuron j ($j = 1$ to M) has an associated vector $\mathbf{t_j} = (t_{j1}, t_{j2}, \ldots, t_{jN})$, that is constituted of the weights t_{ji} on the connections out of neuron j. Such vector $\mathbf{t_j}$ is known as the cluster prototype, that is the internal representation of the category learned by output neuron j. Similarly, there is an input activation vector $\mathbf{w_j}$ corresponding to the weights of connections going from the input layer to output neuron j.

During processing, the input layer receives data inputs and propagates them on the bottom-up connections, which causes activation of neurons on the output layer. The dot product $\cdot$ between input $\mathbf{x_k}$ and bottom-up connections weight vectors determines the activation u_j of each output neuron j:

$$u_j = \mathbf{x_k} \cdot \mathbf{w_j} \tag{1}$$

Competitive selection takes place between output neurons. The winner selected is the neuron $j*$ with maximum activation $j* = \arg\max(u_j)$. The cluster represented by this output neuron is deemed to be the one with the greatest correlation with the input. The input is attributed to the winning output neuron and prototypes weights are updated as such:

$$\mathbf{t'_{j*}} = \mathbf{t_{j*}} \wedge \mathbf{x_k} \tag{2}$$

The prototype weight update with a logical AND guarantees a unidirectional movement of prototypes (monotonically

decreasing magnitude) and thus also contributes to stability (Moore 1988).

To stabilize, the network must iterate through the data until there is no further change of the input assignments to category nodes. It was demonstrated that ART converges to a stable representation after at most $N-1$ presentations of the N data items (Georgiopoulos et al. 1990).

The serial algorithm implementing the binary adaptive resonance theory concepts is as follows:

0. Provide parameter values L and ρ:

   $0 < \rho \leq 1$
   $L > 1$

1. Initialize weights w_{ij} and t_{ji}:

   $w_{ij} = 1/(1+N)$
   $t_{ji} = 1$

2. Present next datum $\mathbf{x_k}$ to network. If no more datum, re-process all data until prototypes do not change.
3. Compute output value for all output neurons:

   $u_j = \mathbf{x_k} \cdot \mathbf{w_j}$

4. Choose most active neuron $j*$ as winner; go to step 5. If there is no active neuron, create a new one j_{new} and initialize its weights :

   $\mathbf{t_{j_{new}}} = \mathbf{x_k}$
   $\mathbf{w_{j_{new}}} = 1/(1+N)$
   then go to step 2.

5. Propagate prototype $\mathbf{t_{j*}}$ of winning neuron to input layer and perform vigilance test:

   $(\mathbf{x_k} \cdot \mathbf{t_{j*}})/||\mathbf{x_k}|| \geq \rho$
   where $||\mathbf{x_k}|| = \sum_{i=1}^{N}(x_i)$ for a N-dimensional datum $\mathbf{x_k} = (x_1, x_2, \ldots, x_N)$;
   If true, go to step 6 (resonance mode);
   Otherwise, go to step 8 (search mode).

6. Update weights :

   $\mathbf{t'_{j*}} = \mathbf{t_{j*}} \wedge \mathbf{x_k}$
   $\mathbf{w'_{j*}} = L(\mathbf{x_k} \wedge \mathbf{t_{j*}})/(L-1+(\mathbf{x_k} \cdot \mathbf{t_{j*}}))$

7. Re-activate all output neurons and go to step 2.
8. De-activate neuron $j*$ and go to step 4.

2.2 The stability problem

Definition 4 (*Assignment or attribution*) Let $\mathbf{x_k}$ be a datum that has reached resonance with prototype $\mathbf{t}_\alpha$ as per step 6 of the ART1 algorithm. An assignment (attribution) $\in$ determines the membership of datum $\mathbf{x_k}$ in cluster α: $\mathbf{x_k} \in \alpha$. We also say that $\mathbf{x_k}$ is assigned or attributed to α.

The ART stabilization is an iterative process that looks at the whole data set up to $R-1$ times or until data elements stop moving between categories. We will henceforth refer to these iterations in the data set as the *stabilization iterations*.

Stabilization works as follows:

- First, assume that a datum $\mathbf{x_k}$ has just been processed by the neural network and is coded by prototype $\mathbf{t}_\alpha$. This is to say that $\mathbf{x_k}$ has been assigned to a cluster (or category node) α of data represented by prototype $\mathbf{t}_\alpha$.
- Secondly, some of the further data processed by the ART1 neural net may also be assigned to this same cluster α and consequently prototype $\mathbf{t}_\alpha$ will be updated to reflect the intersection of all assigned data as per formulae (2).
- Third, entering a stabilization iteration, datum $\mathbf{x_k}$ is presented again to the network and may not anymore be deemed similar enough to prototype $\mathbf{t}_\alpha$. This is possible because the prototype may have been changed by other data. The network must then reassign datum $\mathbf{x_k}$ to another cluster $\mathbf{t}_\beta$.

Thus, the ART network forgets some of its previous experiences during stabilization to re-code assignments of previously processed data to new clusters. In other words, during stabilization, data is *moved between concepts*. When this movement stops, stabilization is achieved.

Stabilization is similar to sleep in living organisms, a period during which experiences of the day are re-processed and properly coded and re-coded in memory. For an artificial learning system such as ART used in a real world, high-volume, 24/7 operation, stabilization may have to occur during system idle time. The various iterations may not occur immediately one after the other as there may be more urgent tasks required, such as processing newly arrived data and delivering it to a user.

For instance, suppose the system under consideration is one that routes, based on topics, intelligence and operational documents to various military analysts. This information is highly perishable and must be processed with high priority, before any further stabilization iteration can continue. As demonstrated in Massey (2005b), one might setup the stabilization to occur during low activity periods on document batches of various sizes. However, that approach was shown

to significantly lower clustering quality and is therefore not acceptable.

The fundamental question one must then ask to clearly delineate the problem with ART1 stabilization is: what happens in between stabilization iterations with data awaiting stabilization? During the first processing pass, data will be assigned to some clusters. Then during stabilization, data will be moved, defeating the whole purpose of providing a stable and consistent environment to users. That is, a datum $\mathbf{x_k}$ has been initially assigned to cluster α. $\mathbf{x_k}$ may be an important document attributed to topic α. The users expect to always find this document under that same topic folder once it has been saved there the first time. However, later on the document may be moved to another topic as stabilization continues. Users will not find the document under the same topic. This movement of data between categories can happen several times and is clearly a problematic situation.

In fact, the whole idea of stabilization rests on the premise that convergence to the so-called stable representation is achieved after the ART network has been able to iterate through the whole data set several times. If a finite data set is being processed, then indeed a stable categorization of the data can be attained.

However, in many real-life streaming data problems, data is for all practical purposes infinite, in the sense that it continues to be delivered to some categorization system until the data source is terminated. This will most likely span years, and therefore even though the data is not really infinite, it is relative to the system and its users who do not see an end to it. Hence, data needs to be processed incrementally for the whole life of the system. Although there may be temporary system shut downs (for example, due to maintenance or upgrades), the flow of data is merely suspended.

Stabilization iterations can be scheduled during low activity periods or following shut downs, but when the system is restarted, novel data will continue to be delivered to the system and will continue to trigger the formation of new categories. Further stabilization iterations will have to include previously assigned data objects. That is, data elements will continue to move, possibly indefinitely since new and more representative prototypes will continue to be created.

In this context, the conditions for neither type of stabilities are met: there is no stable attribution of data to categories and there is endless proliferation of categories. The endless proliferation of categories is not problematic: it is on the contrary a necessary characteristic of a true on-line, incremental, streaming data, plastic learning application. On the other hand, the fact that a datum cannot be assigned to a category permanently is the nature itself of the problem we are addressing here.

Indeed, from a practical standpoint and particularly from a usability engineering point of view, it is absolutely necessary that once data elements have been attributed to a category they are not moved elsewhere. The consequences of multiple movements of data on human users is confusion and loss of productivity due to continual search for information that just keeps moving.

2.3 Conceptual duplication

The solution we propose in this paper is to treat stabilization not as "conceptual shifts" (i.e. data moving between concepts) but rather as "conceptual copies" (i.e. data being duplicated across concepts). We call this stabilization approach *conceptual duplications*. The idea of conceptual duplication for ART1 neural networks modifies stabilization in such a way that all associations between data and categories are remembered by the network, even those that would be invalidated by traditional stabilization. In other words, once a datum has been attributed to a cluster, the network remembers this association. The overall algorithm for ART1 and the neural architecture itself are not changed. The changes can be localized outside the network and consist in remembering all assignments of data elements to categories.

The memory in which conceptual attributions are stored is not part of the ART1 neural network structure itself. We do not claim neurological plausibility; we take an engineering approach to solving a practical problem. Hence, a data structure called *assignment table* is created in regular computer memory (i.e. not part of the neural network structure) and used to accumulate the various categories *assigned* to data elements.

The assignment table has the format shown below, where each column represents a data element and each row corresponds to a stabilization iteration, with cell (i, j) of the table containing the cluster or category number for datum j at iteration i. For example, at iteration 2, datum 5 is assigned to cluster (or category) 8. The first row and column are shown for convenience and contain respectively the datum number (from 1 to R) and the iteration number (from 1 to 4 in this specific example case).

	1	2	3	4	5	6	...	R
1	1	2	2	1	3	4	...	49
2	15	4	4	1	8	4	...	18
3	15	9	4	1	15	7	...	25
4	15	9	4	1	15	7	...	25

The stable state can be observed by considering the last two rows of the assignment table. Indeed, one observes that assigned category numbers do not change anymore for any of the R data elements between iterations 3 and 4. This is an indication that stability has been achieved and that further stabilization iterations will not affect clustering results any further. In this example, we take the conventional view that there is a finite set of R data elements to illustrate the regular stabilization. The same assignment table structure can

be used to accumulate category assignments for an infinite data set and reach stability with the conceptual duplication strategy.

To stabilize with the conceptual duplication strategy, the category each datum is assigned to must be stored in the assignment table following each stabilization iteration. One of the difference compared with the finite data presented in the assignment table above is that one must assume there is enough memory to contain all assignments over the life of the system. This is a purely practical consideration that can be dealt with adequate hardware and memory management. Another difference is that the last row is never truly a last row, since stabilization iterations continue indefinitely. Finally, the last row in the table structure shown above displayed a stable set of assignments: in the infinite data case with conceptual duplication, the last row does not necessarily show a stable attribution of clusters to data but merely a snap shot of the current state of clusters attributions. Although an apparently stable last row (i.e. duplicating exactly the previous one) can occur, it would be purely coincidental stability that would soon be destroyed by the next stabilization iteration incorporating new data.

The stable state in this framework is the one obtained by accumulating all assignments in the table, thus ensuring the data can always be found where it was previously assigned. Therefore, the essential point we make here is that since all assignments are preserved, data effectively ceases to be moved between categories. Hence, data is always present where it was at first for the convenience of user attempting to retrieve a known data. This is the new nature of stability with conceptual duplication. In fact, there is now *duplication* of assignment information leading to a soft clustering solution and thus increased access points to information for users.

However, it may not be desirable in some situations to remember every single assignment, so we introduce a parameter called *evidence*. Evidence is a positive integer value that specifies how many times a category has to be attributed to a datum before it is deemed worthy to remember. With evidence = 1, all assignments are preserved, which is just as if there was no evidence parameter.

Thus, it is important to note that evidence is not a necessary condition for stable learning, on the contrary. Evidence is an optional feature in the conceptual duplication framework that is provided for two reasons: first, a technical consideration as a way to cut down memory usage, and second, for what may be psychological plausibility in what amounts to remembering only those events that carry the most importance, i.e. in our case, that re-occur sufficiently often.

The stabilization process under the conceptual duplication strategy with evidence thus becomes one of first accumulating assignments in the assignment table and second of *trimming* those assignments that do not meet the evidence threshold. Trimming occurs when a change of assignment happens, that is when a new category is attributed to a datum. At that point, it is guaranteed that previous category assignments that did not meet the required evidence level will not be considered again. This is an inherent consequence of ART1 neural networks behavior originating from the prototype weights updates. In effect, the prototypes erode continually because of the intersection of attributed data. It is therefore not possible for a datum to go back to a previous prototype.

If the threshold of evidence is reached by none of the clusters, the last attributed category is temporarily set as the proper one for the datum. This corresponds to the conventional stabilization process. In this case, attribution is temporary since as new data is processed, more stabilization iterations will take place and it is possible that future assignments will meet the evidence criteria. This assignment will therefore be deemed a proper assignment until either a category change occurs, at which time it will be removed (trimmed) and replaced by the new category, or until a better assignment (one that meets the evidence criteria) comes along. It is the only situation where some unstable behaviour is possible and the reason for which evidence should be used carefully, or at the very least low values of evidence employed. Indeed, as evidence is increased, it becomes more and more difficult to meet the threshold; high values of evidence therefore serve little purpose but to re-introduce instability.

We now illustrate the ideas of conceptual duplication with evidence, temporary assignment and trimming. For this purpose, we use the following assignment table which is more representative of the processing of an infinite data set than the previous one. For the two examples below, evidence $= 2$.

	1	2	3	4	5	6	...
1	1	2	2	1	3	4	...
2	15	4	4	1	8	4	...
3	15	9	4	1	15	7	...
4	33	9	4	1	46	7	...
...							

For the first example, one observes that datum 1 exceeds the evidence threshold with cluster 15 but not with cluster 1 or cluster 33. Indeed, over the four stabilization iterations, datum 1 is assigned to cluster 15 on two occasions but only once to cluster 1 and 33. In this case, there is too little evidence to claim that cluster 1 and cluster 33 properly represent datum 1 but there is enough evidence to decide that cluster 15 is an appropriate representation. Note that when the cluster assignment was changed from 1 to 15, the assignment of 1 would have been trimmed (i.e. removed from the assignment table) since it cannot possibly meet the evidence criteria again.

As a second example, lets now consider datum 5, which has its cluster assignment changed from 3 to 8 and then to 15

and 46 during stabilization. There is no cluster that meets the evidence threshold of 2, so the last assignment to category 46 is deemed to properly represent datum 5, but only on a temporary basis. Suppose that on iteration 5, cluster 46 is again attributed to datum 5, then its status would change from temporary to permanent. If on the contrary the next assignment is to some other cluster, then 46 would be trimmed and that new cluster chosen to represent datum 5 on a temporary basis.

Soft clustering is a by-product of conceptual duplication as mentioned previously. ART-based clustering, including ART1, results in hard clustering, that is one category assignment per data object. On the contrary, soft clustering allows multiple categories. Soft-clustering is for example very useful in text clustering since multiple topics can be assigned to a document, making them more easily accessible for users. It is actually a more natural way to organize documents than hard clustering since documents are rarely of a single topic according to human classifiers, a phenomenon known as the inter-indexer inconsistency (Cleverdon 1984).

Attempts to make ART networks produce soft-clusters are few, notably the KMART system (Kondadadi and Kozma 2002). KMART is based on fuzzy-ART and rather than choosing only the winning output, all output neurons that pass the vigilance test are deemed to represent the datum. We have implemented this idea in ART1 but found that the network fails to converge with a finite data set due to the creation of an infinite number of clusters, thus failing on the second type of stability mentioned in the introduction. Conceptual duplication not only solves the problem of unattainable stability in infinite data, but it also offers a working means to achieve soft clustering with ART1.

3 Experimental work

3.1 Methodology

To verify the viability of conceptual duplication we have designed an experiment in the domain of text clustering. The task of text clustering consists in grouping textual documents according to their content, where the groups (the clusters or categories) can be regarded as containing documents of similar topics. We have seen previously in Sect. 2.2 the importance of stability in such an environment.

Our experimental strategy is to compare the quality of clustering obtained with the traditional ART1 stabilization scheme with the quality generated with conceptual duplication clustering. Clustering quality is the determinant factor that can decide whether our approach has potential or not, as we already know that a superior clustering is achieved from the point of view of increased stability and of soft-clustering that facilitates information access.

The text classification benchmark dataset known as Reuter-21578 Distribution 1.0 ModApté split (Apte et al. 1994) was used for the experiment. Each document is transformed in the standard vector space model numerical representation (Salton and Lesk 1968). In this model, a document is characterized by a feature set corresponding to the words present in the document. An ordered list of words appearing in the collection is built, from which common stop words such as articles and prepositions are removed. Words can also be stemmed (i.e. transformed into their lexicographic root) but we have not applied stemming in this experiment since previous experiments showed it resulted in lower quality clustering.

Hence, a document d is translated into an N-dimensional binary vector, where N is the number of features (words) used to represent a document. The vector's ith component corresponds to the ith word in the collection. A value of 1 indicates the presence of this word in d while a value of 0 signifies its absence.

Since the resulting vectors are of very high dimensionality, a final pre-processing step is applied to reduce the number of features. To achieve this goal, words occurring in less than 77 documents were removed. This simple feature space dimensionality reduction was judged very effective for text classification (Yang and Pedersen 1997). The value of 77 is the one that resulted in best quality in previous experiments by eliminating as many words as possible without getting 0-vectors. The final vectors were of dimensionality $N = 357$.

The value of the vigilance parameter is incremented successively by fine steps and the quality of clustering is measured for each value of vigilance. This allows for obtaining an overall and complete view of clustering quality, rather than only measuring quality punctually at the pre-determined so-called natural number of clusters (which in our case would be 93).

The minimal number of clusters present in the data can be determined by minimal vigilance (Massey 2002), computed as $\rho_{\min} < 1/N$. We observe clustering results from $\rho_{\min}$ until the vigilance yields more than 200 clusters. This stopping condition was selected because such a large number of clusters would simply result in information overload for a user and therefore not achieve the intended objective of text clustering.

The Reuter-21578 ModApte corpus comes with a training set (9,603 documents) and a test set (3,299 documents), both including a ground truth solution prepared by human experts. The training set is for supervised classification; ART1 performs unsupervised learning so we use exclusively the test set. The test set is pre-processed into the binary vector format described above, and the expected solution as determined by human classifiers is kept aside and will be used only to compute clustering quality.

Clustering quality is evaluated by computing the F_1 external validity of the solution. This manner of computing quality has been used successfully in clustering before Larsen and Aone (1999). With F_1, one compares the clustering solution $C = \{C_i \,|i = 1, 2, \ldots, M\}$ to the ground truth solution $S = \{S_j \,|i = 1, 2, \ldots, M^s\}$, hence measuring the ability of the clustering algorithm to retrieve the solution prepared by human classifiers. M and M^s are respectively the total number of clusters obtained with the clustering algorithm and the number of topics defined in the desired ground truth solution. The clustering solution C is a set of clusters C_i while the desired solution S is a set of topics S_j. Both C_i and S_j are sub-sets of $D = \{d_0, d_1, \ldots, d_R\}$, the set of R documents to cluster. In the case of soft clustering, the clusters S_j are non-mutually exclusive sub-sets of D. Better quality is achieved with higher F_1 values, in the range [0,1]. F_1 is given by

$$F_1 = \frac{\sum_{j=1}^{M^s} |S_j| F_{1j}^*}{\sum_{j=1}^{M^s} |S_j|} \tag{3}$$

F_{1j}^* is the F_1 value of the cluster that best matches topic j i.e., of all clusters, it is the one maximizing its F_1 value with respect to topic j. The F_1 value of a cluster i with respect to a given topic j is

$$F_{1i} = \frac{2\alpha_i}{2\alpha_i + \beta_i + \chi_i} \tag{4}$$

where

$$\alpha_i = |C_i \cap S_j| \tag{5}$$
$$\beta_i = |C_i| \cap \alpha_i \tag{6}$$
$$\chi_i = |S_j| \cap \alpha_i \tag{7}$$

which are respectively the number of true positives, the number of false positives and the number of false negatives.

We note that Eq. 4 is obtained by simple algebraic manipulations from the well-known F_1 effectiveness measure of information retrieval and text classification (Sebastiani 2002; Van Rijsbergen 1979):

$$F_b = \frac{(b^2 + 1)pr}{b^2 p + r} \tag{8}$$

where the precision p and recall r are defined as:

$$p = \alpha/(\alpha + \beta) \tag{9}$$
$$r = \alpha/(\alpha + \chi) \tag{10}$$

Parameter b determines the balance between precision and recall and its value is usually set to 1, which is what we have done to derive Eq. 4. In text classification, the number of true positives α, false positives β and false negatives χ are not computed exactly as in clustering since one has a priori knowledge of which class corresponds to which topic in the ground truth solution. Details of the differences between text classification and text clustering F_1 computation are presented in Massey (2005a).

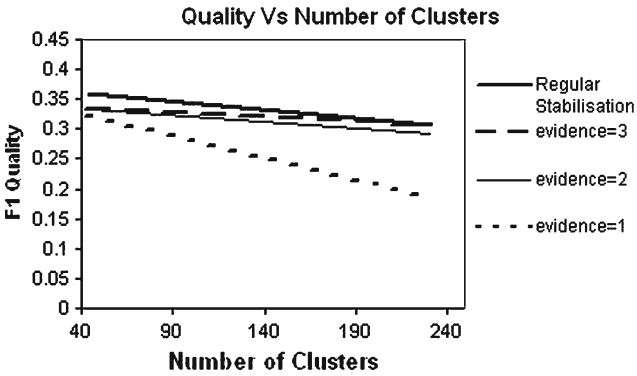

Fig. 2 Quality improves as evidence is increased

3.2 Results and discussion

Fig. 2 shows the experimental results. We have processed the text data over multiple values of vigilance generating between 40 and 230 clusters (x-axis). For each clustering solution thus obtained, the F_1 quality was computed (y-axis). The process was repeated with evidence thresholds of 1, 2 and 3. Since clustering quality varies dramatically between cluster number values, a graph generated on these values would be difficult to read. Therefore, for readability purposes the graph of Fig. 2 displays a linear regression of the original highly variable data points.

We can then observe on Fig. 2 that the F_1 quality improves when the value of evidence is increased from 1 to 3. Hence, the fact that ART1 remembers previous concept attributions based on stricter evidence threshold improves quality. Furthermore, we also observe that compared to regular stabilization available in the standard ART1 model, conceptual duplication with evidence = 3 results is slightly lower quality. Indeed, the average F_1 quality for the three values of evidence = 1, 2 and 3 is respectively 0.28, 0.32 and 0.33. With the regular ART1 stabilization, average F_1 quality is 0.34, which is a meagre 0.01 (3%) higher than conceptual duplication with evidence = 3 and 0.02 higher (6%) higher than with evidence = 2.

We recall that an evidence of x means that a category had to be assigned to a document x times before it was deemed important enough to be remembered during stabilization. A document can then possibly meet the evidence criteria for multiple clusters, thus resulting in a soft-clustering solution. This is the second advantage of conceptual duplication in addition to providing a way of stabilizing under infinite streaming data. In the current experiment, we have used a maximum evidence threshold of 3. Further increases in evidence result in little further gain in quality. In fact, as

one becomes more demanding with evidence, there is less and less opportunities for conceptual duplication to actually occur. Consequently, as evidence increases the solution turns into the usual hard clustering of ART1.

Hence, although quality is not improved by conceptual duplication, the decrease in quality is minimal, particularly in the case of an evidence of 3 for which $F_1 = 0.33$ compared to $F_1 = 0.34$ without conceptual stabilization. Moreover, the two advantages of solving the stabilization problem and of offering a working soft-clustering version of ART1 make the approach very worthy.

A disadvantage of the evidence parameter is the generation of temporary assignments which re-introduce an unstable behavior. Indeed, only evidence = 1 (which effectively amounts to turning off the optional evidence functionality) totally eliminates instability. However, the evidence parameter has the dual advantage of, first from a practical point of view, restraining memory consumption and second, from a semantic aspect, constraining what is deemed to be an acceptable experience worthy of being remembered. Consequently, higher evidence retains those assignments that are of higher quality, as we have observed with the higher clustering quality with evidence = 2 and 3 compared to evidence = 1.

The advantage of saving memory with evidence is relatively secondary and can be handled by memory management schemes. This is an operational issue rather than a theoretical one, thus with no negative impact on the foundation of conceptual duplication. The second problem is rather serious, since it forces a compromise between quality of clustering and stability.

The beauty of conceptual duplication is that it is a general framework that retains and observes the assignment produced by the clustering algorithm, here an ART network. There is no choice or compromise required: conceptual duplication may very well work at both levels (i.e. with and without evidence). First, all attributions are preserved as they are generated (no more trimming) and second, as time passes and evidence can indeed be accumulated, we annotate previous assignments to indicate the best ones (i.e. with the evidence value). In this manner, no forgetting is allowed to occur but the best memories can be annotated as such and re-enforced as more evidence is accumulated. The evidence accumulated on a given category assignment becomes a score on the validity of this category assignment.

Although one may be interested in investigating the severity of unstable behaviour introduced by temporary assignments or in determining an optimal value for evidence, the previous discussion renders such experimentation rather useless other than from a purely academic interest. Indeed, based on our experiments, it has become obvious that evidence is not a parameter of the learning or a threshold that must be met, but rather an output of the process that indicates the goodness of an attribution. Doing so, temporary assignments, trimming and any instability are eliminated. In the end, the general framework of conceptual duplication provides the required mechanisms to attain stability and more. Our experimental results allowed us to re-visit our initial design and improve it.

One area of future work to conduct in a real-life environment is to measure memory usage as time passes and a growing number of data is processed. Trimming, which was partially designed as a mechanism to limit the size of the assignment table is gone. Therefore, all assignments are preserved and the assignments table will potentially consume large amounts of memory.

Another interesting work to conduct is to perform usability testing to determine how easy it is for users to find information with the evidence score and the multiple assignments. One objective of such a study with human users is to uncover how many assignments for each datum is optimal in finding information and the impact of a large number of assignments on the cognitive load of a user.

4 Conclusion

In this paper, we discussed the stabilization problem of data elements moving between categories in a dynamic environment when using ART1. Since there is never a real state of "completeness" in infinite streaming data (i.e. new data continue to be submitted indefinitely to the learning system), ART1 keeps creating new categories infinitely. During ART's stabilization phase, the data are moved between categories until a stable state can be reached. However, the condition for stability can never be reached because the neural network keeps forming new categories based on new data being processed which causes more movement of data.

We have proposed a new stabilization strategy named conceptual duplication to resolve this problem. In this case, the nature of stability has changed radically: from stability that iteratively assigns a single concept attribution onto each datum of a static dataset to a stabilization that handles dynamic, streaming data and accumulates an evidence score to designate the best multiple conceptual attributions. With conceptual duplication, past concept attributions are remembered and scored rather than being forgotten as in the regular ART1 stabilization process.

Our objective and evaluation strategy in this work was to test the viability of conceptual duplication from the point of view of quality compared to the usual ART1 stabilization mechanism. This objective was achieved. Future work will allow us to look at practical issues such as usability issue and memory consumptions.

Experimental results in a text clustering task have shown that as greater evidence is demanded, F_1 quality first increases but then tapers off with higher evidence which forces the

network back into its usual unstable hard clustering behaviour. Stabilization with conceptual duplication results in a F_1 clustering quality that is slightly lower than traditional ART1 stabilization. However, conceptual duplication offers two major advantages: first, it resolves the important problem about data moving between categories during stabilization in a dynamic data environment; and second, it provides a soft-clustering solution which is a very useful addition in many practical applications. These major advantages more than outweigh the small loss in quality.

Our experiments have shown that incrementing evidence results in better quality clustering, but at the cost of re-introducing instability. This observation has allowed us to revise and adjust the detailed mechanism of conceptual duplication and turn evidence into a score that identifies the best category assignments rather than a threshold to keep or reject categories. Hence, both the stability of low evidence and the quality of higher evidence can be obtained.

The conceptual duplication principle is a very useful improvement to ART1. This improvement allows for the utilization of ART1 in a streaming environment with infinite data. Realistic environments often involve infinite datasets, which renders conceptual duplication an even more essential contribution since it makes ART1 usable in such real-life applications.

References

Apte C, Damerau F, Weiss SM (1994) Automated learning of decision rules for text categorization. ACM Trans Inf Syst 12(2):233–251

Carpenter GA, Grossberg S (1995) Adaptive resonance theory (ART). In: Arbib MA (ed) Handbook of brain theory and neural network. MIT Press, Cambridge

Carpenter GA, Streilein WW (1998) ARTMAP-FTR: a neural network for fusion target recognition, with application to sonar classification: AeroSense. In: Proceedings of SPIE's 12th annual symposium on aerospace/defense sensing, simulation, and control, Orlando, April 13–17, 1998

Caudell T, Smith SDG, Johnson C, Wunsch D, Escobedo R (1991) An industrial application of neural networks to reusable design Adaptive Neural Systems. Technical Report BCS-CS-ACS-91-001. The Boeing Company, Seattle

Cleverdon C (1984) Optimizing convenient online access to bibliographic databases. Inf Serv Use 4(1):37–47

Georgiopoulos M, Heileman GL, Huang J (1990) Convergence properties of learning in ART1. Neural Comput 2(4):502–509

Grossberg S (1976) Adaptive pattern classification and universal recording : I. Parallel development and coding of neural feature detectors. Biol Cybern 23:121–134

Kondadadi R, Kozma R (2002) A modified fuzzy art for soft document clustering. In: Proceedings of the international joint conference on neural network. Honolulu, HA

Larsen B, Aone C (1999) Fast and effective text mining using linear-time document clustering. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 16–22

Massey L (2002) Determination of clustering tendency With ART neural networks. In: Proceedings Of recent advances in soft-computing (RASC02), Nottingham, UK, Dec 2002

Massey L (2003) On the quality of ART text clustering. Neural Netw 16(5–6):771–778

Massey L (2005a) An experimental methodology for text clustering. In: Proceedings of 2005 IASTED international conference on computational intelligence (CI 2005), Calgary, Canada, July 4–6, 2005

Massey L (2005b) Real-world text clustering with adaptive resonance theory neural networks. In: Proceedings of 2005 international joint conference on neural network, Montreal, Canada, July 31–August 4, 2005

Moore B (1988) ART and pattern clustering. In: Proceedings of the 1988 Connectionist Models Summer School, pp 174–183

Salton G, Lesk ME (1968) Computer evaluation of indexing and text processing. J ACM 15(1):8–36

Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47

Van Rijsbergen CJ (1979) Information retrieval. Butterworths, London

Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings Of ICML-97, 14th international conference on machine learning, pp 412–420