Trade-off Between Capacity and Generalization in a Model of Memory

Guy Tannenbaum (guy_tannenbaum@yahoo.com) and **Yehezkel Yeshurun (hezy@post.tau.ac.il)**
School of Computer Science, Tel Aviv University
Ramat Aviv, Tel Aviv 69978, Israel

Shimon Edelman (se37@cornell.edu)
Department of Psychology, Cornell University
Ithaca, NY 14853, USA

Abstract

Although computational considerations suggest that a resource-limited memory system may have to trade off capacity for generalization ability, such a trade-off has not been demonstrated in the past. We describe a simple model of memory that exhibits this trade-off and describe its performance in a variety of tasks.

Keywords: memory; computational model; capacity; generalization; trade-off.

Introduction

Because the probability of a cognitive agent encountering precisely the same stimulus twice is infinitesimally small, memories of past experiences are only useful for guiding future behavior insofar as they can be generalized so as to apply to new variations on familiar themes. Intuitively, it would seem that in a memory system better generalization would have to come at the expense of reduced capacity. Indeed, the famous mnemonist patient studied by Luria (1968), whose memory capacity seemed practically unlimited, was oblivious even to simple patterns in the memorized items.

From the functional standpoint, this intuition can be related to the distinction commonly made between "simple" or episodic memory (pertaining to statistically rare but important events, such as who did what to whom), where capacity is the key goal, and conceptual memory (pertaining to statistically redundant patterns of events in the environment), where generalization between "similar" items is crucial (Merker, 2004). Because these memory functions, along with the many others in the human brain (Rolls, 2000), are subject to certain constraints on the available resources, one expects memory systems to exhibit a trade-off between capacity and generalization.

Surprisingly little computational work has been devoted to testing this prediction. Does the expected trade-off arise generically, in any resource-limited memory system? Despite important early insights into the computational underpinnings both of simple and of conceptual memory (Brindley, 1969; Marr, 1969, 1970), subsequent "connectionist" memory models, such as those of Hopfield (1982) or Kanerva (1988), did not consider this question, in part because they had not been intended to provide generalization capabilities. At the same time, the more comprehensive theoretical frameworks for the understanding of memory, such as that of Minsky (1985), have typically been only partially implemented (Hearn, 2001). An explicit computational model designed to provide both storage and generalization has been recently developed by Mueller and Shiffrin, but its reported evaluation (Mueller, 2006; Mueller & Shiffrin, 2006) seems to be qualitative rather than quantitative.

Our goal in the present study has been to investigate the emergence of a trade-off between capacity and generalization under conditions that are as general as possible. To that end, we chose to focus on implementing a functional (rather than neuromorphic) computational model of pattern storage and generalization, which extends that of Moll and Miikkulainen (1997). This allows us to relate our results to existing methods and findings regarding memory capacity.

The computational framework

Our model operates on vectors of positive integers. Its building blocks are "neurons" with real-valued activation, connected via real-valued synaptic weights. Performance is measured as a function of the resources available to the system, and of how they are allocated between the different aspects of each task.

Capacity

We measure memory capacity by assessing the model's ability to recognize previously encountered patterns. Given a query pattern, the model returns the probability of having encountered it before. The results are plotted in the form of a receiver operating characteristic (ROC). Capacity is then defined as the number of patterns that the model can memorize while maintaining a given area under the ROC curve. This metric has a natural psychological interpretation: human memory is often faced with the task of deciding whether or not a pattern has been seen before (Koriat, Goldsmith, & Pansky, 2000). Alternatively, capacity can be defined in terms of completion of partial patterns (Moll & Miikkulainen, 1997).

Generalization

Capturing an artificial hierarchical taxonomy. A basic task that involves generalization is similarity estimation: the memory representation of some structured collection of objects, such as items drawn from a taxo-

nomic tree, should capture the various relative similarities of these objects. We presented the model with patterns representing items taken from a three-level strictly hierarchical taxonomy (objects within sub-classes within classes) and compared the pairwise similarities between their memory traces to the true similarities, defined by the shortest paths between the corresponding nodes in the tree (Tenenbaum, Griffiths, & Kemp, 2006).

Capturing a lexical taxonomy. To obtain another perspective on its generalization ability, we used the model to derive a hierarchical clustering of lexical items from a natural language text corpus (Finch & Chater, 1991). This procedure begins by enumerating the unique words in the corpus. The model is then presented with patterns formed by sliding a window along the text, each pattern consisting of the list of word tags in the window. Finally, model's representations of words are clustered according to their pairwise similarities (for this, it must be possible to assess similarity between partially specified inputs).

The model

The convergence-zone episodic memory of Moll and Miikkulainen (1997), of which the present model is an extension, consists of two layers of real-valued units (the feature map layer and the binding layer) and bidirectional binary connections between the layers. Initially all connections between the binding layer and feature map layer are inactive and have the value of zero. A pattern is stored in the memory in three steps: (i) those units that represent the appropriate feature values of the pattern are activated; (ii) a subset of m binding units are randomly selected in the binding layer to to encode this pattern; (iii) the weights of all the connections between the active units in the feature maps and the active units in the binding layer are set to 1.

To complete a partial pattern, the corresponding feature maps units are activated. The activation propagates to the binding layer through all connections that have been turned on so far. At this stage units in the binding layer can have different activity levels. The activity level of all the units connected to all the active feature units is the number of the active feature units. Other binding layer units are connected only to a subset of the active feature units, and will therefore have a lower activity level. Only those binding layer units with the maximal activity level are retained, and the others are turned off. The activation of the remaining binding units is then propagated back to the feature maps. A number of units are activated at various levels in each feature map, and again, only the most active unit in each feature map is retained, resulting in a complete unambiguous pattern.

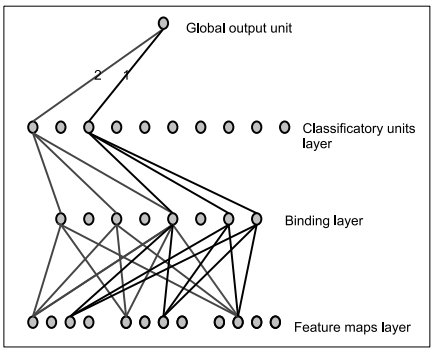

Figure 1: The architecture of the memory model. The feature maps layer, the binding layer, and the connections between them encode information regarding which unique patterns have been encountered by the model. The classificatory units layer and its connections to the binding layer and global output unit encode the number of times a particular pattern has been encountered. The model in this illustration has encountered two instances of the pattern $\{1, 1, 2\}$ and one instance of $\{3, 3, 2\}$.

Handling input statistics

Our version of the model adds three new functions: handling repeating input patterns, answering old/new recognition queries, and maintaining and reporting the statistics of the encountered patterns. In particular, when queried with a full pattern, our model simply returns the frequency with which this pattern has been encountered; when queried with a partial pattern, the model returns the marginal frequency with which such partial pattern had occurred in the input (e.g., the response to $\{3, 4, *, 7, *\}$ is the frequency of encountering a pattern in which the first, second, and fourth feature maps have the values of 3, 4 and 7, and the two remaining ones are "don't cares"). The ability to handle queries about the statistics of encountered patterns can be useful, for example, when dealing with patterns representing location, food availability, and the year's season: the present model can directly support decision making about optimal foraging strategies. In addition, handling input statistics can help the model achieve generalization, as explained shortly.

The architecture of the model

Compared to that of Moll and Miikkulainen (1997), the present model has two new layers (see Figure 1). The first is the classificatory layer, each of whose units represents a different pattern; its connections to the binding layer are binary. A classificatory unit becomes active only when all the binding layer nodes it is connected to are activated. The second new layer has a single global output unit, whose connections to the classificatory units are real-valued. Its output is set to the sum of all the strengths of the connections to the currently

active classificatory units, normalized by the sum of all the strengths of the connections to all the classificatory units. Initially, all the connections between the binding and classificatory units and between the classificatory units and the global output unit are inactive.

Storing patterns

When a pattern is presented to the model to be stored, the activity propagates from the feature map layer onwards. In the binding layer, only those units with maximal activity remain active. The activity than propagates to the classificatory layer, where a unit is activated only if all the binding units it is connected to are. Finally, the activity reaches the global output unit which sums the strengths of all its incoming connections.

If the output unit is not activated at this stage, the pattern is considered novel. A random set of m binding units and one new classificatory unit are allocated for representing it. The global output unit is connected to the new classificatory unit with an initial strength of 1 (Figure 2).

If, on the contrary, the global output unit is activated by the initial feedforward sweep, the pattern is considered familiar. The strength of the connections between the global output unit and any active classificatory units is increased by 1. Unless the model is overloaded, there is only one such active classificatory unit: the one allocated when the present pattern was first encountered (Figure 3).

Answering queries

When the query is a complete pattern, and if an identical pattern has been stored, a unique classificatory unit will be activated and the activity of the global output unit will be proportional to the frequency of this pattern (out of all the patterns presented). If no such pattern has been stored, no classificatory unit will be activated, and the output strength will be 0 (unless the model is overloaded, which may lead to errors).

When the query is a partial pattern, the binding nodes that become active are the ones that participate in representing patterns in which the specified features have the given values. Similarly, among the classificatory units, which respond only when all the relevant binding nodes are, only the nodes representing patterns in which the relevant features have the given values are activated. Summing the connections strengths of these units to the global output unit has the effect of calculating the marginal distribution of the specified feature values.

Old/new queries are processed by first allowing the activity to propagate to the global output node. If it remains inactive, the model responds with 0 (meaning that the pattern has definitely not been seen before). If the global output node is active, the response to the query is $output = 1 - ((b_a - m)/b_t)$, where b_a is the number of

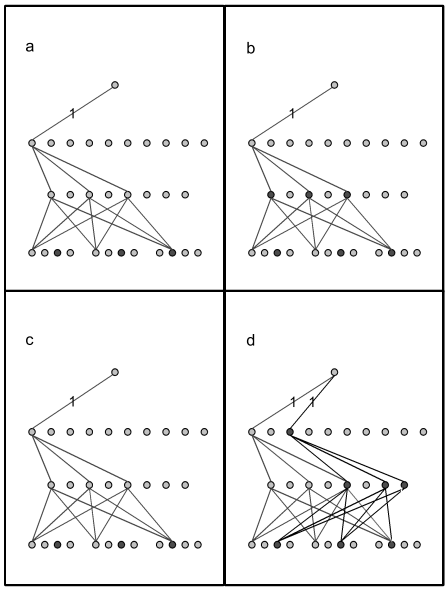

Figure 2: (a) The model, which has stored the pattern $\{1, 1, 2\}$, is presented with the pattern $\{3, 3, 2\}$. (b) Activity propagates to the binding layer. (c) Because none of the nodes have activity level of three (the number of specified values in the input), none of them remain active, and the classificatory nodes and global output node remain inactive. (d) Three random binding nodes and a new classificatory node are recruited for representing this pattern.

active binding nodes, m is the number of binding nodes recruited when storing a new pattern, and b_t is the total number of binding nodes. The rationale behind this response is that the more binding nodes are currently active, the more likely it is that the m binding nodes representing some random pattern will be activated. As the load on the model increases, so does the number of binding layer to feature layer connections, resulting in lower certainty when answering this type of query. This calculation can also be used for estimating when the model is about to become overloaded and should not be used for storing more patterns.

Resources

The present model implements more functions, but also uses more resources (nodes and connections), than the original convergence zone model (it needs one classificatory node per unique pattern, making the capacity less than the total number of nodes). It may be instructive to compare the model to one in which there is no binding layer and the feature maps are connected directly to the classificatory units. Such a model would use fewer nodes, and would not suffer from the errors generated due to the probabilistic process of recruiting

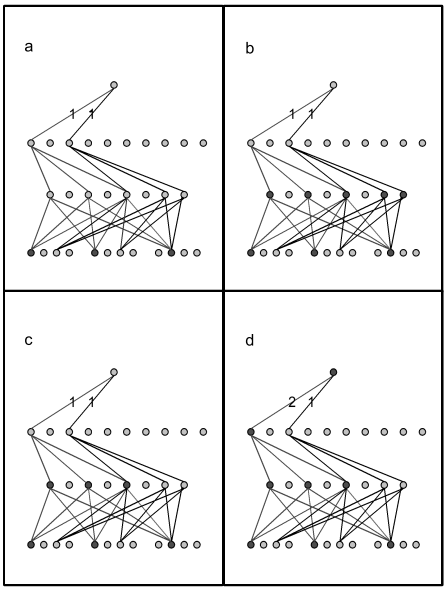

Figure 3: (a) The model, which has previously stored the patterns $\{1, 1, 2\}$ and $\{3, 3, 2\}$, is presented with $\{1, 1, 2\}$ for the second time. (b) Activation propagates to the binding layer. (c) Only those binding layer nodes with an activity level of three remain active. (d) Activation propagates to the classificatory layer and the global output unit and the strength of the connection between the active classificatory node and the global output node is increased.

binding nodes. However, when considering the number of connections required by these models, its disadvantage becomes clear. In our version of the model, the number of connections is $c_1 = f \times b + b \times c + c = b \times (f + c) + c$, where f is the number of feature layer nodes, b the number of binding nodes, and c the number of classificatory nodes. In comparison, in the model without the binding layer, the number of connections would be $c_2 = f \times c + c$. Therefore, when the number of binding nodes is smaller than half of both the number of feature nodes and the number of classificatory nodes (a reasonable constraint), the total number of connections in our version of the model is smaller. In effect, manipulating the size of the binding layer allows controlling the number of connections needed by the model, at the expense of tolerating more errors.

Comparing patterns

Algorithm 1 (see next page) is used for calculating the similarity between two patterns. The co-occurrence statistics needed for calculating the surprise factor in line 9 of the algorithm are estimated by querying the model with the corresponding partial patterns (i.e., let $n = 5$, $i = 1$, $k = 2$, $v_i^j = 4$, $v_k = 5$; the frequencies of the following partial patterns would then be queried: $\{4, 5, *, *, *\}, \{4, *, *, *, *\}, \{*, 5, *, *, *\}$.)

A straightforward neuronal implementation of this algorithm can be based on maintaining multiple copies of the model which all learn the same patterns. Each of these can be hardwired to output one of the required co-occurrence statistics when answering a pattern comparison query. As we show later, not all the possible co-occurrence statistics are required to achieve reasonable performance. Using a randomly selected subset of the co-occurrence statistics is likely to provide good performance, as long as the subset is large enough.[1] This approach leads to a trade-off between the number of copies of the model (which limits the number of co-occurrence statistics used for similarity judgments) and the amount of resources per copy (which limits the capacity). If using more the statistical information results in better generalization (which happens to be the case, up to a point; see Figure 6), then this trade-off leads in turn to the generalization vs. capacity dilemma.[2]

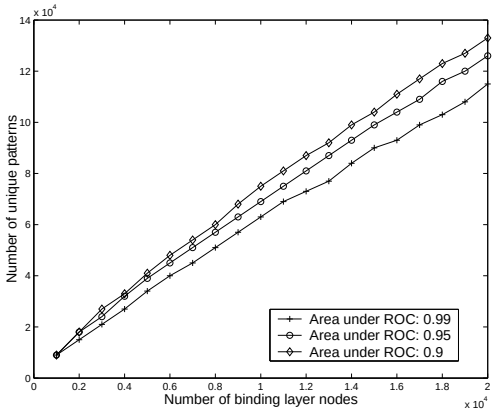

Figure 4: Memory capacity vs. the number of binding nodes (10 feature maps of size 50, averaged over 3 runs). Capacity is defined as the maximum number of patterns that can be stored while still maintaining a given area under the ROC curve.

Simulation results

Basic performance characteristics. The model's capacity and performance under increasing load are plotted in Figures 4 and 5. The model is assessed on two different tasks: differentiating between old and new patterns and recalling the pattern frequencies. Up to a certain

[1] In this case line 7 of algorithm 1 needs to be changed to only traverse some subset of all the possible feature map values.

[2] One of the many issues not addressed in this work is the effect of projecting the pattern space into a feature space which can support better similarity judgements between pairs of patterns. The trade-off identified above is expected to arise regardless of the feature selection and similarity judgment generation methods used, as long as increasing the amount of resources at their disposal leads to better performance.

Algorithm 1 Calculating the similarity between a pair of patterns

Input: a pair of patterns $p1 = \{v_1^1, v_2^1, ...v_n^1\}$ and $p2 = \{v_1^2, v_2^2, ...v_n^2\}$
Output: the similarity between the pair of patterns s
for $i = 1 : n$ **do** {*n is the number of feature maps*}
  for $j = 1 : 2$ **do**
    for $k = 1 : n$ **do**
      if $k \neq i$ **then**
        for all feature map values $m = 1 : M$ **do** {*M is the number of feature map values*}
          Calculate the surprise factor f of encountering a pattern in which $v_i = v_i^j$ and $v_k = m$.
          $f \leftarrow \frac{Prob(v_i = v_i^j \wedge v_k = m)}{Prob(v_i = v_i^j) \times Prob(v_k = m)}$
        end for
      end if
    end for
    Concatenate the results of these computations to form a vector d_j with the length of $M \times (n-1)$
  end for
  $c_i \leftarrow$ correlation between d_1 and d_2.
end for
$s \leftarrow mean(c_i)$ {The correlation scores could instead be weighted by variance, as in the Mahalanobis metric.}

load, performance in both tasks remains good and the error is almost constant. After this point, performance degrades rapidly. Figure 6 shows generalization performance as a function of the resources allocated to this task.

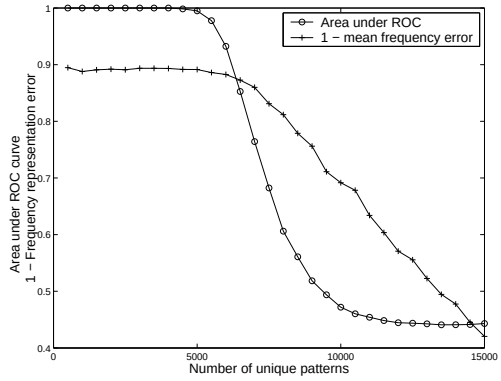

Figure 5: (∘), performance (the area under the ROC curve) vs. load (stored number of patterns). The model was given an increasing number of patterns to store, while being tested on differentiating between stored and unseen patterns. (+), normalized mean error in reporting pattern frequency vs. the load.

Trade-off between capacity and generalization. Designing a memory system with multiple instances of the model (as suggested earlier) while keeping the total number of binding nodes constant necessitates a decision: how to allocate the nodes among the instances of the model. Having a small number of copies would allow each one to have a large binding layer and therefore high capacity, but at the cost of being forced to use fewer co-occurrence statistics when comparing feature map values, leading in turn to degraded generalization performance. Figure 7 depicts the resulting trade-off between capacity and generalization. It combines the data used to generate Figures 4 and 6. The abscissa values covary with the resource trade-off: the same $300,000$ binding nodes are divided into 300 instances on the left, and into only 10 on the right.

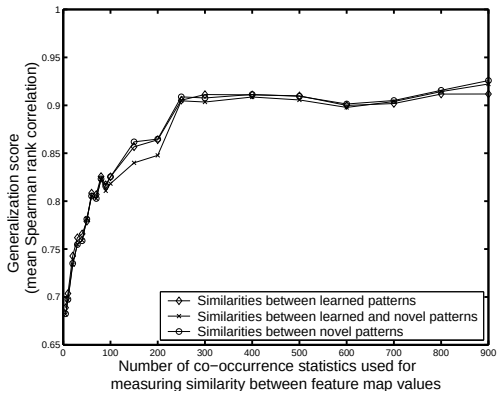

Figure 6: Generalization vs. the size of the random subset of the co-occurrence statistics used for calculating pattern similarity (4 feature maps of size 400; 100 binding nodes; averaged over 10 runs). (⋄), Spearman rank correlation between true tree distances and the similarities between stored patters. (×), same, for pairs containing one stored and one new pattern. (∘), same, for pairs of new patterns. See Algorithm 1 for details.

Lexical taxonomy. The dendrogram in Figure 8 depicts the contextual similarities among the 500 most frequent words in Lewis Carroll's *Alice in Wonderland*, as distilled by our model. The results are similar to those reported by Finch and Chater (1991).

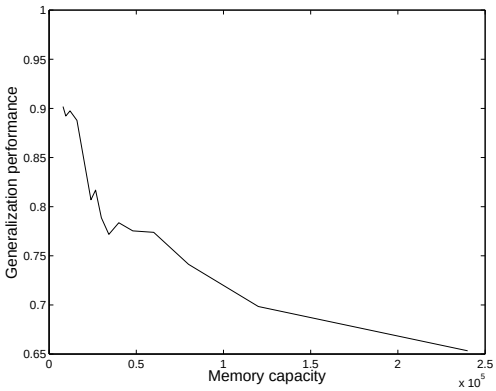

Figure 7: Trade-off between memory capacity and generalization (see text for explanation).

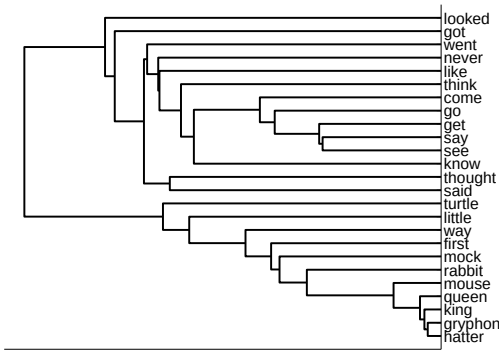

Figure 8: Hierarchical clustering of words from *Alice in Wonderland*. A memory model with 20,000 binding nodes was presented with patterns generated by moving a sliding window of length 5 over the text. The model was then queried for similarities between all pairs of words. Clustering was performed according to these similarities.

Summary

Human memory is characterized by high capacity, context sensitivity, and access flexibility. Computational models of memory need to quantify these properties and explicate the relationships between them. The present work explored one such relationship: between the capacity of a memory system and its ability to generalize among similar items. The trade-off that we had expected and were able to demonstrate in a simple, almost generic memory model provides a useful perspective on how memory works.

The computational model presented here lacks sophistication and neurobiological realism, yet it is a step in the right direction, because it is capable not only of storing and recalling patterns, but also of making certain generalizations about the stored items. Future work in this direction would have to address the need for a well-founded approach to statistical inference on the part of the model, ideally thus bringing it in line with the modern Bayesian framework for cognition (Chater, Tenenbaum, & Yuille, 2006), and also the need to test it against the body of behavioral and neurobiological findings concerning human memory.

References

Brindley, G. (1969). Nerve net models of plausible size that perform many simple learning tasks. *Proc R Soc Lond B Biol Sci.*, *174(35)*, 173-191.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*, 287-291.

Finch, S., & Chater, N. (1991). A hybrid approach to the automatic learning of linguistic categories. *Artif. Intell. and Simul. Behav. Qtrly.*, *78*, 16-24.

Hearn, R. (2001). *Building grounded abstractions for artificial intelligence programming.* Msc thesis, Massachusetts Institute of Technology.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.*, *79*, 2554-2558.

Kanerva, P. (1988). *Sparse distributed memory.* Cambridge, MA: MIT Press.

Koriat, A., Goldsmith, M., & Pansky, A. (2000). Toward a psychology of memory accuracy. *Annual Review of Psychology*, *51*, 483-539.

Luria, A. (1968). *The mind of a mnemonist.* Cambridge, MA: Harvard University Press.

Marr, D. (1969). A theory of cerebellar cortex. *J. Physiol.*, *202*, 437-470.

Marr, D. (1970). A theory for cerebral neocortex. *Proceedings of the Royal Society of London B*, *176*, 161-234.

Merker, B. (2004). Cortex, countercurrent context, and dimensional integration of lifetime memory. *Cortex*, *40*, 559-576.

Minsky, M. (1985). *The Society of Mind.* New York: Simon and Schuster.

Moll, M., & Miikkulainen, R. (1997). Convergence-zone episodic memory: Analysis and simulations. *Neural Networks*, *10*, 1017-1036.

Mueller, S. T. (2006). REM-II: A Bayesian model of the organization of semantic and episodic memory systems. In *Proc. Cognitive Neuroscience Society Meeting.*

Mueller, S. T., & Shiffrin, R. M. (2006). REM II: A model of the developmental co-evolution of episodic memory and semantic knowledge. In *Proc. Intl. Conference on Learning and Development (ICDL).*

Rolls, E. T. (2000). Memory systems in the brain. *Annual Review of Psychology*, *51*, 599-630.

Tenenbaum, J. B., Griffiths, T. L., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, *10*, 309-318.