Reconsolidating Data Structures

Thomas Heinis[†] [*], Anastasia Ailamaki[‡]

[†] *Department of Computing, Imperial College, London, United Kingdom*

[‡] *Data-Intensive Applications and Systems Lab, École Polytechnique Fédérale de Lausanne, Switzerland*

ABSTRACT

For decades forgetting has been treated as an abnormality, a malfunction of the brain that leads humans to lose stored information. Recent results, however, suggest that forgetting is not only a malfunction of the human storage system, but also a useful feature. In order to guarantee a quick response in the face of the limited processing power of the brain, acting quickly on less or reduced information is key.

With storage becoming ever cheaper and continually growing it has become standard practice today to store each and every single data item. However, even increasingly powerful processors cannot deal with this data deluge. In this paper we consequently argue that forgetting and its mechanisms should be a part of today's data management, particularly for techniques requiring fast and/or approximate query answers. While forgetting or shedding information may have far-reaching implications for current methods in data management, in this paper we focus on discussing forgetting (and learning) in the context of data synopses.

1. INTRODUCTION

Experimental evidence in neuroscience research recently revealed that human forgetting [25] is not only a side effect of disease or of age, nor do we forget because the capacity of the brain is limited. Rather, we forget because the brain has limited computational power, yet we live in an environment where we need to make rapid judgments [1, 3]. The brain has consequently evolved to forget so that we can react quickly. In a world where we constantly absorb and learn information, we need to forget in order to enable split second decisions: we would rather react quickly based on imprecise or approximate information rather than too late. In an approximate world, who needs precision anyway?

One might argue, that forgetting is important in an environment with limited computational power and where approximate answers are sufficient, but that it has no place in a world with almost unlimited computational power. There are, however, many scenarios where computational power is limited or where response time is restricted. Forgetting can therefore also work to our advantage and in this paper we argue that forgetting or shedding information can also be beneficial in data management.

Forgetting may have broad implications and may trigger interesting discussions. Recent work, for example, introduces the notion of data rotting [18], i.e., a mechanism to periodically reduce data in a database to avoid it growing boundless (while ideally also adding the removed data in condensed and curated form back to the database). In this paper, however, we focus on the exact mechanisms of forgetting or, put in simple terms, fading memories (as well as learning) and use them to develop reconsolidating data structures, i.e., data structures that manage information through "forgetting" but also "learning". While we initially discuss the idea of reconsolidation structures broadly, we also elaborate on its application to data synopses in more detail. Surely, the case for data synopses [10] for quick and approximated answers has long been made, but what we discuss here is how to use some of the brains mechanisms to efficiently manage data synopses along with the resulting data management research challenges.

The remainder of this paper is structured as follows. We first discuss the neuroscience background of forgetting and learning in Section 2 and discuss where these mechanisms can fit into data management in Section 3. In Section 4 we elaborate on how the mechanisms of forgetting, i.e., reconsolidation, can be a powerful design principle when managing data synopses and discuss the resulting research challenges in Section 5. We discuss other potential applications in Section 6 and conclude in Section 7.

2. NEUROSCIENCE BACKGROUND

Several theories have been developed in recent decades to explain forgetting (as well as learning). For decades, the cognitive neuroscience theories of memory decay [4] (memory traces fade over time) and memory interference [16] (memory traces encoded with similar stimulus replace each other - similar to collisions in a hash map) were popular explanations for forgetting. More recent theories of consolidation [19] and reconsolidation [2], however, have gained much support lately, primarily due to their sound explanation of underlying cellular and molecular mechanisms [24]. They are today the most widely accepted explanations for forgetting (and learning).

In the following we discuss and summarise the consolidation/reconsolidation theories and discuss their implications.

[*]This work was done while the author was at EPFL.

10.5441/002/edbt.2015.66

2.1 Learning

The basic premise of learning is that memory traces [23] are not immediately stable or permanently stored when they are learned (encoded). Instead, memory traces are made permanent in the consolidation phase through strengthening (or weakening) the connections (synapses) between the neurons involved in a memory trace. By strengthening connections (or a synapses) between them, signals can be relayed quicker between two neurons and the firing patterns corresponding to memory traces can consequently be adapted.

The underlying biochemical process to strengthen synapses is very slow and can take several hours, during which memories are not stable. During this consolidation phase, the synapses need to be repeatedly stimulated. The hippocampus therefore repeatedly replays the firing patterns corresponding to the traces.

Any stress on the brain in general can interrupt the consolidation phase and memory traces may never become permanent in the brain or may only become weak. Anecdotal evidence for this interruption, for example, is that learned facts cannot easily be recalled after an all-nighter of learning (and implied lack of sleep). Similarly, interrupting consolidation with heavy drinking tends to erase whatever was recently experienced leading to a mental blackout.

2.2 Forgetting & Updating Memory

Although for a considerable time forgetting was assumed to be governed by completely different mechanisms, it is surprisingly similar to learning [2]. When retrieving memory traces (remembering), the traces become unstable or labile. The same synapses that were strengthened during the consolidation phase are rendered unstable and need to be reconsolidated in a process lasting for several hours similar to consolidation (despite built on different biochemical mechanisms).

Whilst retrieving information as a memory, the associated trace is therefore temporarily unstable and can be altered. Similar to interference during the consolidation process, also interference during reconsolidation alters the trace. Depending on the interference, whether it is positive or negative, memory can become stronger or weaker than it initially was. Reconsolidation has been extensively studied in fear memory, for example, in the experimental treatment of post-traumatic stress disorder, memories are recalled to make them unstable and interference in the form of electric stimulation is used to erase them. Similarly, positive feedback can be used to reinforce memory or to alter it [11]. Using positive feedback repeatedly can make memory traces more precise or can alter them substantially.

Experiments further show that the temporal dynamics of memory reconsolidation depend on the strength and age of the memory [20], such that younger and weaker memories are more easily reconsolidated than older and stronger memories. Similarly, the question whether old memory traces are updated or stored as new primarily depends on the age of the memory and also on the similarity of the event when recalling the trace: the similarity needs to be greater for updating an old memory, whereas for only recently encoded memory, the similarity threshold can be considerably smaller [9].

2.3 Reconsolidation in the Context of Data Management

To summarise from a computer science (or data management) perspective, forgetting is not simply a linear function of time. Instead, it is either a consequence of stress/disease or, surprisingly and crucially, from remembering it. Recalling memory makes the information unstable and requires reconsolidation. The reconsolidation process can lead to both, improvement (through updating) or to degradation (withering) if the reconsolidation process is interrupted. If and how much the memory is improved or degraded also depends on cognitive cues, i.e., feedback, during reconsolidation: positive feedback leads to improved memory whereas negative feedback generally leads to forgetting [11].

A key aspect of forgetting or learning memory is that traces are not completely erased (or completely accurately stored), i.e., they are not erased or learned as one, but are instead gradually degraded or improved. Thus, the mechanism of reconsolidation is a powerful means to reduce the amount of information that needs to be taken into account to answer a question (compute a query result) while still ensuring that relevant information is retained.

Today's ever-cheaper storage hardware allows storing nearly everything and eliminates the need of deleting old data. Similar to the brain, however, our ability to store data far exceeds today's data processing capacity and so, to keep answering queries quickly, we need to radically reduce the data taken into consideration. Using a mechanism similar to forgetting (and reconsolidation) in today's data management could be very powerful to gradually delete data permanently.

A more cautious approach to use the powerful mechanism of forgetting (and learning) in data management, however, does not permanently delete information/data. Instead it can be used to manage summaries of data by removing (forgetting) from summaries and adding data (from the full dataset) to them. The key aspect of forgetting (and learning) is how memory fades away (or is strengthened) gradually. This mechanism enables reducing the space needed to store a data structure by reducing its precision or expanding its size by increasing the precision. Consequently, its size and therewith query time cannot only be controlled by dropping or adding single items as a whole, but by reducing their size individually; similar to fading memories.

Using reconsolidation on proxy data structures has similarities to using caches in data management (and computer science in general) but also differs substantially.

First, caches are primarily small because they are expensive as opposed to the brain where the size is limited to guarantee response time. In the brain, forgetting is primarily driven by the need to reduce the time to process information. In the cache, however, the idea is rather to reduce communication time and the data is therefore moved to faster storage hardware (and also closer to processing). Essentially, it is not the processing time that is reduced, but rather the communication time, which helps in turn to reduce the overall processing time.

Second, in the case of reconsolidation, recently used data items are labile or prone to change whereas in caches, items that have not been accessed in a while are evicted and replaced by frequently and recently accessed data items (depending on the caching policy).

Third and crucially, items in a cache are typically either completely loaded or completely evicted (if room is needed for more items) and cache management is completely oblivious of the content of the items cached. Reconsolidation, on the other hand, is aware of the items' contents and reduces or improves their precision or resolution gradually.

With its ability to fade and strengthen data/information, reconsolidation is consequently particularly relevant for applications (or data structures) that allow for imprecision. This process is similar to data synopses [7], which can give quick approximate answers instead of accessing all data to give an excruciatingly slow (but precise) answer. Data synopses are pivotal today because the same technological advances that enable large-scale analysis of data also enable the generation of data on a similar scale. Yet the growing data generation capacity combined with cheap storage technology is likely to keep on outpacing the analysis capacity.

3. RECONSOLIDATING DATA STRUCTURES

What we contemplate here are reconsolidation data structures that, similar to caches or data synopses, act as a surrogate for the full dataset. Similar to the brain, they are limited by a time constraint to answer a query (and by giving a time constraint, they are also constrained by the space that they can use because of limited processing power available) and manage the precision of the data summary. The reconsolidation data structure, an approximated data summary, is used to quickly answer the query approximately.

At the core of reconsolidating data structure is the idea of treating data items as not oblivious to their content, but rather as aware of it. The major difference to existing caching strategies (and data synopses) is that data items are not dropped based on a binary decision, but their precision is decreased or increased, depending on how much the user is interested in them. Gradually improving or degrading the precision of data can of course only be tolerated in applications where precision is not crucial but where time matters. This is mostly true for applications where, similar to the brain, an approximate but quick answer is more important than an exact and slow one like, for example, in applications where humans consume the result.

To design reconsolidating data structures, we map the neuroscience mechanisms of forgetting and learning on the idea of data synopses and caches. This essentially means that we focus on a reconsolidating data structure (RDS) that acts as a surrogate for a complete (and potentially massive) dataset and is used to answer user queries. We further interpret queries to an RDS as a memory retrieval, i.e., recall of a memory trace. We then define accurate and complete queries to the underlying data structure dataset as new learning (or as interference with existing memory traces), i.e., improving precision of the information. Absence of a query to a dataset, on the other hand, is interpreted as if the answer from an RDS (and with it the RDS) is precise enough or, more precisely, too accurate and therefore uses more space than needed and its accuracy can thus be degraded. The age of memories in RDS is used to decide how fast the information is degraded: similar to the brain, old memories are degraded slower than new memories.

Put more simply: a query to the reconsolidation data structure answers the query and makes the data items touched labile/unstable. A subsequent query to the full data set means that the approximate query result was not precise enough and consequently learning starts, i.e., the precision of the data items touched is increased. Absence of a query to the data means that the approximate result was precise enough or, more importantly, too precise and the precision of the data items touched is reduced.

Of course application specific cues on the quality of the result can also be used to decide whether to improve or degrade precision, similar to Google's result ranking application where a click on a result is fed back to Google and is used to rank the results for the same keywords in future searches. Any such cue, however, is particular to an application and cannot be used in general.

Using a reconsolidation strategy for managing data synopses or caches, however, bears the risk that items are loaded into the surrogate dataset that are retrieved once but never again. Their precision may therefore never degrade and they may never be entirely dropped from the dataset. They will remain in the dataset indefinitely taking up space. However, we primarily propose reconsolidation data structures so that the response time in which approximate query answers are given is limited (and not to strictly adhere to a space budget). Still, if space is also a key concern, then additional mechanisms (like memory decay [4] where memory degrades as a function of time) can be used.

Clearly, for this idea to work in practice, the difficulty is to define the process of reducing or improving precision and to define how to query the resulting data structures. The latter, however, will in most cases remain the same. We will discuss this in the following for a number of applications where reconsolidation can prove useful.

4. EXAMPLE APPLICATION: DATA SYNOPSIS

The reconsolidation data structures we propose and discuss here are a rather abstract concept/mechanism that can be applied to different types of data structures. In the following we discuss how they can be used in the context of data synopses to make them a powerful tool in the face of a mass of ever growing data. We first provide background information on data synopses and then discuss the potential of reconsolidation for synopses.

4.1 Synopses Background

Research in data synopses has in the past primarily been driven by applications like data streams and cardinality estimation for query processing [7]. Their very nature, however, presents a great opportunity to accelerate approximate query execution in the context of big data.

4.1.1 Overview

The basic idea of data synopses is that the full data set is summarised, typically by using compression [7]. The synopsis acts as a surrogate of the data and is queried instead of the full dataset. Through compression or summarisation the synopsis is usually considerably smaller than the full dataset itself and, consequently, queries are executed substantially faster on the synopsis. The execution time of a query on the synopsis depends primarily on the size of the synopsis (unless additional auxiliary data structures like indexes are used). The size of the synopsis in turn depends on dataset characteristics, i.e., how easily compressible the data is, and is further controlled by the compression used.

Because of its substantial compression ratio, lossy compression is often used but doing so also leads to imprecise representations of the data. Data synopses based on lossy compression can consequently only approximately answer queries. Clearly there is a trade-off between size of the data synopsis (and thus query execution time) as well as the quality of the approximation, i.e., the smaller the approximation, the less accurate it is and thus the bigger the error becomes. Irregardless of the quality of approximation used, the key of data synopses is that they provide a user with tight error bounds expressing how accurate the received query result is.

Data synopses have in the past primarily been studied and used in the context of data streams and to estimate the cardinality of database tables (for query planning) [7]. With data growing beyond what can be today handled efficiently and reasonably, data synopses are again being considered as an interesting and competitive approach: instead of analysing the potential terabytes and petabytes of data in big data applications in a time consuming process, substantially smaller synopses can be queried almost instantly.

4.1.2 Types of Synopses

Considerable effort in the past has primarily developed four types of synopses. First, random sampling [21] takes samples at random out of the dataset (or the relation) and is very well suited for aggregate queries. Samples can be taken online, at query time from the full dataset, or offline, i.e., prior to querying to store them in a synopsis data structure. Online sampling is particularly interesting to improve the quality of the query result continuously: as long as the user is willing to wait, samples can be taken to improve the accuracy of the approximate query answer. For massive data that are primarily stored on disk, however, taking the samples online from the full dataset, as the query is being executed, is unlikely to be feasible due to the high cost of random access to the disk. Instead, big data sampling has to take samples offline, i.e., once from the full dataset and store the samples separately. Clearly, in the case of offline sampling, the more samples that are taken, the more precise the approximation, but the bigger the synopsis will also be.

A second well-researched type of synopsis is histograms [14]. In the context of databases, histograms play a crucial role in query optimisers and are often used for the purpose of data visualisation. Histograms summarise the data into bins each with its own value range, e.g., each bin stores the count of values/tuples in its range. Doing so makes them particularly useful for range-count queries, but they also have the potential to be used for general analysis queries [7].

Synopses based on wavelets summarise and approximate the data through wavelets [5]. Essentially, wavelet transformation is applied to relations or to time series resulting in a collection of wavelet coefficients. The size of the synopsis depends on how many coefficients are stored, which in turn defines the accuracy with which queries can be answered. The size of the synopsis alone, however, does not define the query execution time: at runtime, query execution can choose to ignore coefficients, thereby reducing query execution time, but also the degree of precision.

Relatively new are synopses based on sketches [6]. The basic rationale is to summarise the data per query type. As opposed to sampling, all data is considered, but only a small summary is retained (e.g., for a sum query all values are added up and only the sum is stored). As each query can be supported by a sketch, this approach is very powerful and applicable to all types of queries. Defining a new sketch per query type, however, requires considerable effort.

Artificial neural networks are also used to learn or approximate datasets [22] (for example time series [8]). Inspired by neuroscience, they use a graph with neurons as vertices and synapses as edges to answer queries approximately. Originating from machine learning neural networks are, however, rarely used as synopses in the management of data.

4.2 Reconsolidating Data Synopses

The basic idea of using reconsolidation data structures as a data synopsis [7] (or put more simply, using a synopsis featuring reconsolidation) is to improve or degrade the precision of the synopsis depending on the queries. The precision of regions frequently queried (in both, the synopsis and the dataset) is increased and the precision of those regions that are only queried in the synopsis is reduced.

A straightforward target to apply the idea of forgetting (or reconsolidation in general in order to support improving memory) is to use it on data synopses based on neural networks. They are modelled very similarly to the brain by using a graph with neurons as vertices and the connections (edges) between vertices represent the synapses. Synapse strength (and thus ultimately the encoding of information) can be modelled as weight of the edges. Degrading the synopsis is accomplished by reducing the weight of the edges (or by removing them altogether) and increasing the precision by increasing the weights. Changing the weight of the edges, however, will not effectively reduce the size of the synopsis. Hence, although neural networks lend themselves perfectly to the idea of forgetting, forgetting does not have a significant impact on the size of synopses based on neural networks or the time to execute queries.

There is, however, no need to restrict the idea of data synopsis or models based on neural networks. Much more interesting to apply reconsolidation to are data synopses based on, for example, wavelets [5]. Wavelets are used to approximately interpolate and therefore compress the underlying data set. Clearly the data synopsis is only an approximation of the real data set, but by using reconsolidation, areas or ranges of interest, i.e., queried over in the synopsis and in the complete dataset, can be stored with more precision. Others, retrieved only from the synopsis, can gradually be degraded by using less precise wavelets for interpolation.

Similarly, in data synopses based on histograms [10], precision can also be increased locally for interesting regions and can be decreased for uninteresting ones. In either case, the queries can be executed as usual on the reconsolidating synopsis. In any scenario where data synopses provide approximate answers, error bounds or guarantees are crucial.

5. DATA MANAGEMENT RESEARCH CHALLENGES

Applying the idea of reconsolidation to data synopses introduces several interesting data management challenges and thus research opportunities.

5.1 Adapting Data Synopsis Resolution

Key to the idea of reconsolidation data structures is to change the precision in given areas of it. Queries to the synopsis and the full datasets are used to infer what ranges (e.g., areas in a spatial model) the user is interested in exploring and analysing. The precision is increased in areas where the scientist is interested in and decreased elsewhere as a result of users' queries.

For sampling, assuming the samples are stored in a synopsis and are not taken online (in which case there is no data structure other than the full dataset needed), this can be achieved by taking and storing more samples from areas that users are interested in and deleting samples from areas where the interest is low.

In the case of histograms, improving precision is accomplished by adding bins and, thus, making the intervals (or value range of the bins) smaller and consequently more precise. Conversely, reducing the precision is similarly straightforward: neighbouring bins can be combined efficiently to make the resolution coarser (by increasing the value range) in areas with little interest. Figure 1 illustrates with the initial histogram (top) and the reconsolidated histogram (bottom) where in ranges of little interest (e.g., 1-30 & 70-100) bins are collapsed and in areas of a lot of interest bins are split.

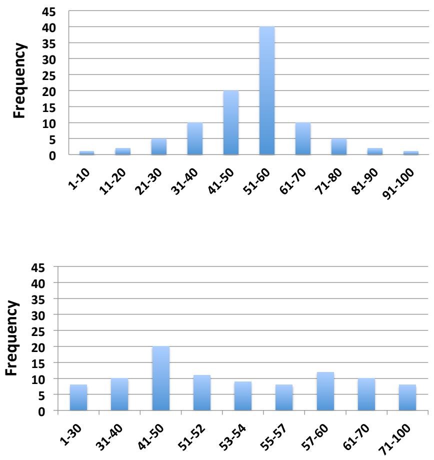

Figure 1: Initial histogram (top) and reconsolidated hisogram with adjusted precision (bottom).

For both, histograms as well as sampling, improving or reducing the precision is not very challenging. For wavelets and sketches, the two types of synopses that enable answers to more general classes of queries, however, changing the precision is not straightforward.

Wavelets are an interesting type of synopsis for reconsolidation. By definition, querying a wavelet based synopsis can be accelerated by ignoring coefficients that provide more precise query answers, thereby reducing the precision on the fly. Doing so, however, does not reduce the size of the synopsis itself and to apply the principle of reconsolidation, we can drop coefficients, in case there is little or no interest in an area, or learn and add coefficients (albeit in a computationally intense process from the full dataset) to the synopsis.

In the case of sketches, the precision is difficult to adjust. The challenge for sketches is that they are very application specific and it is thus difficult to find a generic way to define/implement their reconsolidation.

Adapting the resolution is consequently particularly challenging for wavelets and sketches where research has yet to develop efficient (for wavelets) and generic (for sketches) means to adapt to the precision. Research not only has to develop mechanisms to adapt the precision but also determine the methods to decide the exact area as well as the new level of precision.

5.2 Error Bounds for Variable Resolution

A key aspect of synopses is the idea to answer queries only approximately, but with tight error bounds. Providing the error bounds for query answers that only touch areas with the same precision is straightforward.

Given a synopsis with variable resolution, however, makes it challenging to compute the error bounds. Assume, for example, a synopsis based on histograms where the intervals in certain value ranges are smaller than in others, i.e., the interval length is variable. Clearly, ranges with smaller intervals have higher precision and thus smaller errors (and vice versa). The question, however, is how do we combine the different errors into a meaningful and intuitive error bound for a query that touches intervals of variable length?

A crucial research question, thus, is how to compute the error bounds based on a synopsis with variable precision, i.e., how to make the variable precision quantifiable or how to turn it into error bounds. Novel methods have to be developed to quantify the error bounds in case the ranges with varying precision are used.

5.3 Feedback Mechanisms

The feedback mechanisms described for data synopsis so far work by monitoring access to the full dataset. If the full dataset needs to be accessed, then the result (or the error bound) provided was not precise enough and consequently we have to learn, i.e., increase the resolution of the synopsis by taking more samples from the full dataset. If we do not have to access the full dataset, we can forget, i.e., the precision is decreased.

One research challenge consequently is whether better feedback mechanisms can be found. Clearly, for many applications better solutions can be used, e.g., using user input. Any such approach, however, is application specific and the research question is if generic mechanisms can be found to decide whether to learn or forget.

5.4 Answering Queries

The basic idea is for the user to gain as precise an answer as required from the data synopsis. If the answer, however, is not precise enough, then the full dataset is queried to provide a sufficiently precise answer (and also the data synopsis is improved through learning). The challenge to be addressed thus is how we can manage to only read the information additionally needed from the full dataset. This may be rather straightforward for wavelets, since in their case only additional coefficients can be read from disk (if they are stored on disk along with the full data) to make the result more precise.

For all other types of synopses the question of how to efficiently complement synopsis data with results, i.e., how to retrieve the minimal amount of information needed from disk and combine the results efficiently, is a challenging research question.

5.5 Data Organization

Changing the precision of the synopsis means either adding or removing data from it. A crucial research question directly affecting the performance of querying the synopsis is how to organise the data (presumably, given its size) on disk. Simply appending data when learning will lead to a data structure that requires excessive random disk access while only removing information (without reorganising the structure) means that considerable unnecessary data will be read and so the question becomes how can we design an updated efficient data structure for synopses?

6. OTHER APPLICATIONS

Reconsolidation has applications beyond data synopses and can be used in applications where imprecision can be tolerated or where data is imprecise/uncertain by nature.

6.1 Reconsolidating Caches

Clearly reconsolidation makes little sense if the data items in a cache are very small. In the case of hardware caches [13] (e.g., CPU caches) it is therefore unlikely to be used. Still for other types of caches in applications that can tolerate imprecision, reconsolidating can be used.

6.2 Content Distribution Networks

An interesting application for reconsolidation is caches in the context of content distribution networks [15, 17]. Objects in these caches are typically big and are consumed by users. A straightforward application is the degradation or improvement of image quality (or other multimedia content like videos or music) whether it is to return the image directly to the user or to query over it.

Degrading the image quality can be achieved by reducing the resolution, the size or by restricting the colour palette. Any of these approaches will reduce the effective size of the objects and thus query execution on the reconsolidation structure is accelerated. Imaging formats based on bitmaps or rasters may be difficult to degrade or improve precision easily and quickly, but there also exist layered formats where layers can be added or dropped individually.

7. CONCLUSIONS

What we present here is the need for forgetting and its mechanisms in the brain. We argue that forgetting should also have its place in data management. The brain, however, deals with imprecise information while many data management applications require precision and it is therefore not obvious where and how forgetting fits into data management. Given that storage is becoming ever cheaper, there seems to be little need to delete at all.

What we argue here, however, is that there is still a need to delete (or forget) to ensure timely query answers. This is particularly important as the quantity of data is growing quicker than the CPUs are becoming faster. Given the comparatively slow CPUs, we have to shed information to guarantee answers within a given time [12].

What we propose here is the mechanism or the design principle of reconsolidation that should be used in the design of applications and data structures. As we show with its application to data synopses, forgetting can be a powerful mechanism for managing data and yet entails considerable research challenges.

By discussing the example of data synopses we also demonstrate how the compelling mechanism of reconsolidation can be applied to applications where imprecision is acceptable.

Maybe the power of this approach is not so much in mapping the principle of reconsolidation strictly onto data management. However, we believe that reconsolidation (increasing or degrading precision) is a powerful mechanism, particularly for the management of data synopses and caches, which are becoming increasingly important to guarantee quick answers in face of today's (and tomorrow's) data deluge.

8. ACKNOWLEDGEMENTS

This work is supported by the Hasler Foundation (Smart World Project No. 11031) and the FET Flagship the Human Brain Project (grant agreement number 604102).

References

[1] J. Barrett and K. J. Zollman. The Role of Forgetting in the Evolution and Learning of Language. *Journal of Experimental Theoretical Artificial Intelligence*, 21(4):293–309, 2009.

[2] A. Besnard, J. Caboche, and S. Laroche. Reconsolidation of Memory: A Decade of Debate. *Progress in Neurobiology*, 99(1):61 – 80, 2012.

[3] R. A. Bjork and A. S. Benjamin. Successful Remembering and Successful Forgetting : a Festschrift in Honor of Robert A. Bjork, 2011.

[4] J. Brown. Some Tests of the Decay Theory of Immediate Memory. *Quarterly Journal of Experimental Psychology*, 10(1):12–21, 1958.

[5] K. Chakrabarti, M. N. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00.

[6] G. Cormode and M. Garofalakis. Sketching Streams Through the Net: Distributed Approximate Query Tracking. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05.

[7] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine. Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches. *Foundations and Trends in Databases*, 4(1):1–294.

[8] G. Dorffner. Neural Networks for Time Series Processing. *Neural Network World*, 6:447–468, 1996.

[9] S. J. Gershman, A. Radulescu, K. A. Norman, and Y. Niv. Statistical Computations Underlying the Dynamics of Memory Updating. *PLoS Computational Biology*, 2014.

[10] P. B. Gibbons and Y. Matias. New Sampling-based Summary Statistics for Improving Approximate Query Answers. In *Proceedings of SIGMOD '98.*

[11] M. J. Hays, N. Kornell, and R. A. Bjork. The Costs and Benefits of Providing Feedback During Learning. *Psychonomic Bulletin Review*, 17(6):797–801, 2010.

[12] T. Heinis. Data analysis: Approximation aids handling of Big Data. *Nature*, 515(7526):198, 2014.

[13] J. L. Hennessy and D. A. Patterson. *Computer Architecture, Fourth Edition: A Quantitative Approach.* Morgan Kaufmann Publishers Inc., 2006.

[14] Y. Ioannidis. The History of Histograms (Abridged). In *Proceedings of the 29th International Conference on Very Large Data Bases*, VLDB '03.

[15] K. Johnson, J. Carr, M. Day, and M. Kaashoek. The Measured Performance of Content Distribution Networks. *Computer Communications*, 24(2), 2001.

[16] J. Jonides and D. Nee. Brain Mechanisms of Proactive Interference in Working Memory. *Neuroscience*, 139(1):181 – 193, 2006.

[17] J. Kangasharju, J. Roberts, and K. W. Ross. Object Replication Strategies in Content Distribution Networks. *Computer Communications*, 25(4), 2002.

[18] M. Kersten. Big data space fungus. In *Conference on Innovative Data Systems Research (CIDR '15).*

[19] J. L. McGaugh. Memory – a Century of Consolidation. *Science*, 287(5451):248–251, 2000.

[20] K. Nader. A Single Standard for Memory; the Case for Reconsolidation. *Debates in Neuroscience*, 1(1), 2007.

[21] G. Piatetsky-Shapiro and C. Connell. Accurate Estimation of the Number of Tuples Satisfying a Condition. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data.*

[22] D. F. Specht. A General Regression Neural Network. *IEEE Transactions on Neural Networks*, 2(6), 1991.

[23] L. Squire. Mechanisms of Memory. *Science*, 232(4758):1612–1619, 1986.

[24] N. C. Tronson and J. R. Taylor. Molecular Mechanisms of Memory Reconsolidation. *Nature Reviews Neuroscience*, (4):262 – 275, 2007.

[25] J. T. Wixted. The Psychology and Neuroscience of Forgetting. *Annual Review of Psychology*, 55(1):235–269, 2004.