

Thinking Fast and Slow: A CBR Perspective

Srashti Kaurav¹

cs19s013@smail.iitm.ac.in

Devi Ganesan¹

gdevi@cse.iitm.ac.in

Deepak P^{1,2}

deepaksp@acm.org

Sutanu Chakraborti¹

sutanuc@cse.iitm.ac.in

¹Indian Institute of Technology Madras, Chennai-600036, India

²Queen's University Belfast, United Kingdom

Abstract

In a path-breaking work, Kahneman characterized human cognition as a result of two modes of operation, Fast Thinking and Slow Thinking. Fast thinking involves quick, intuitive decision making and slow thinking is deliberative conscious reasoning. In this paper, for the first time, we draw parallels between this dichotomous model of human cognition and decision making in Case-based Reasoning (CBR). We observe that fast thinking can be operationalized computationally as the fast decision making by a trained machine learning model, or a parsimonious CBR system that uses few attributes. On the other hand, a full-fledged CBR system may be seen as similar to the slow thinking process. We operationalize such computational models of fast and slow thinking and switching strategies, as Models 1 and 2. Further, we explore the adaptation process in CBR as a slow thinking manifestation, leading to Model 3. Through an extensive set of experiments on real-world datasets, we show that such realizations of fast and slow thinking are useful in practice, leading to improved accuracies in decision-making tasks.

1 Introduction

Case-based Reasoning (CBR) is a method of solving a new problem by reusing solutions from similar previous experiences, which are recorded as cases. Given a target problem, the system *retrieves* similar cases and *reuses* their solutions by adapting them to suit the needs of the target problem. The solution thus obtained is *revised*, often using human intervention, and an informed decision is taken as to whether the solved problem should be *retained* as part of the case base. Schank (1982) proposed the conceptualization of dynamic memory in order to facilitate story understanding, which inspired the field of CBR. However, CBR, as it has been deployed in real world settings, often seems to have lost sight of the cognitive goals that inspired its origin.

Designing cognitively inspired systems is one of the long-term goals of Artificial Intelligence. In order to progress in this direction, we need to have some understanding of human behaviour and cognition. In his book “Thinking Fast and Slow” (Kahneman 2011), Daniel Kahneman explains that human cognition has two aspects to it – “fast thinking”

and “slow thinking.” Fast thinking operates quickly, with little or no effort and no sense of voluntary control, whereas slow thinking is associated with effortful mental activities that require concentration and complex computations. Most of the normal regular tasks or events we are habitual of performing in our daily lives can be done easily using fast thinking. But events which are unusual or surprising slow us down and demand slow thinking.

In terms of CBR, fast thinking is used for processing a normative or general query which can be solved using some generalized knowledge. For solving queries that are different from the norm or are surprising, we may need some relevant and specific localized knowledge (Kolodner 2014). Figuring out that specific knowledge may slow down the retrieval process. For example, let us consider a CBR system designed to aid in diagnosing and fixing car problems. The system should be able to reason about the normal working of a car or some usual defects, even without being aware of the details about the car’s architecture. In contrast, for a car with a radically different make and design, the generalized knowledge may not be helpful for reasoning. The competence of a CBR system in solving problems is closely related to the idea of *alignment* (Massie, Craw, and Wiratunga 2006), the extent to which similar problems tend to have similar solutions. Given a certain representational choice, it is typical for certain regions of the problem space to be better aligned than others. Problem solving is fast when CBR system responds to a query from regions of high case base alignment. A CBR system slows down when it attempts to solve a query from regions of low case base alignment where retrieved nearest neighbours don’t have similar solutions, and adaptation is complex and computationally demanding (Craw and Aamodt 2018). In this paper, we present several novel ways of operationalizing the fast-slow thinking dichotomy by using different decision-making models for each paradigm. Using a fast thinking module for normal processing and switching to slow thinking as needed, can substantially reduce average computation time. There are several potential applications in time-constrained domains that can benefit from such realizations. An example is in the context of autonomous navigation where a robot must be quick in responding to obstacles that are very close, but can afford higher response times for obstacles that are far away (Grubb and Bagnell 2012). In such settings, it is usual to

trade off accuracy for speed; in the face of an impending urgency, a quick and dirty solution may come in handy. One of the goals of the empirical studies reported in this work is to establish that such tradeoffs can be realized in the CBR context, and thus we expect such designs to be useful in practice.

The approaches proposed in this work address two important questions. Firstly, we examine three different configurations to model fast and slow problem solving, both while restricting ourselves to design choices within the CBR context, as well as in settings where eager Machine Learning approaches complement CBR systems effectively. Secondly, we propose mechanisms that can initiate the process of switching from fast to slow thinking in each of these three configurations.

2 Background

Two Modes of Thinking: In a widely acclaimed work (Kahneman 2011), Daniel Kahneman explains that our thought process results from two types of thinking. One is intuitive and immediate, called fast thinking, and the second is slow thinking, which involves active conscious reasoning. Kahneman also calls these thinking modalities as *System 1* (fast) and *System 2* (slow). These refer to just two conceptual models of cognitive operation and do not imply any physiological or biological separation within the brain.

Distinctions: Upon seeing the image in Figure 1, one can understand that the lady is angry and is about to say something loudly. The conclusion drawn by looking at the image was automatic and thus attributable to System 1. This inference stems from intuition rather than as an output of a reasoning process. In fact, it may even be hard to explain verbally as to how the conclusion was reached. The reasoning may be the product of both the mix of evolutionary knowledge en-



Figure 1:
Example

grained in the brain, and observational knowledge accumulated over years of interacting with humans that get codified as intuitions. This is Fast Thinking, we are not usually aware of how or when it happens. We draw an analogy between this and what happens in machine learning and deep learning models which are statistical models trained over historical data that can make quick decisions for a new input. Another analogy is to consider decision making based on a *few attributes* within a CBR process as fast thinking. In contrast to fast thinking, consider solving a mathematical task, such as 17×24 ; the answer to this does not immediately come to mind. However, the process of solving this mathematical task is very clear in mind, and most of us would be capable of explaining it step by step. The step by step, deliberative process, which can be explained consciously, is the function of System 2. In our analogy, we relate such slow thinking with the CBR process that involves identifying similar cases for a new problem based on a well-rounded similarity assessment, developing ways of reusing the similar cases, and eventually solving the new problem with knowledge distilled from the similar cases.

Complementarity: System 1 is capable of fast decision making, but could be less accurate in scenarios where the intuitions have not developed so much. These could involve unfamiliar situations such as interpreting complex body language from humans in a different cultural background. While it would be considered rational to fall back to the deliberative and detailed processing in System 2 upon encountering unfamiliar situations, Kahneman observes that humans tend to replace the problem with a simpler and more familiar problem, leading to a cognitive bias called the *availability heuristic*. In our analogy, this relates to engaging statistical ML and deep learning models even in spaces where they are not competent (e.g., training data is sparse). Much like the alternative to the cognitive bias is to deliberately engage System 2, our analogy suggests that ML models should similarly switch to CBR's deliberative model using a well-rounded similarity assessment.

Dual Operation: Humans have the ability of metacognition (Metcalfe, Shimamura, and others 1994), which enables them to *reason about their thinking processes*. If used effectively, this could be well-suited to operationalize *mode switching* accurately; in other words, towards deciding when to switch to System 2. Consider a chess player who encounters a familiar game situation, where it is cognitively easier to rely on System 1 to make a reflex move. Upon encountering an unfamiliar or overly sophisticated case which she can reason as being something that her intuition is not good at, the chess player may need to do a reasonable cost-benefit analysis of each possible next move through engaging System 2. The *availability heuristic* and *metacognition* are often in tension. The former involves substitution and is cognitively easier since it only engages System 1, whereas the latter would lead to a slow but likely more appropriate response. Humans are not always experts at accurate mode switching, due to this tension. Expert chess players often rely on intuitions even where they don't have enough intuition capability, and end up making game losing moves even though their System 2 knew better.

Realizing the Duality Computationally: As outlined earlier, we look to model System 1 as either a trained ML model or as a parsimonious CBR system that uses a few attributes. On the other hand, System 2 is a CBR system that uses a well-rounded similarity heuristic that can holistically assess the similarities between cases. These yield two dual-system configurations by pairing the full-fledged CBR with either of the two System 1 options. We consider various models of mode switching as well. These include external feedback to System 1 decisions; for example, this is the hypothetical chess playing situation where the player always uses System 1, but is offered a second chance (to engage System 2) whenever the decision ends in a mishap (e.g., loss of an important chess piece). The other options are various ways of realizing metacognition or expertise assessment. These include ways of assessing confidence (which is suited to ML models) as well as those of assessing the competence of the two Systems in the immediate vicinity. The next section will describe our various computational realizations of the dual system operation.

3 Proposed Methods

As outlined above, we propose multiple methods for computational realization of the dual system operation. We explain the three frameworks in detail below.

3.1 Model 1: ML/DL & Full CBR

Fast thinking or System 1 in human cognition works by way of using knowledge embedded in the brain structure (through evolutionary processes) or intuitions accumulated within it over the course of life (through past observations and inferences). In Model 1, we use the latter to draw an analogy with trained statistical models in machine learning (including deep learning frameworks), where abundant past historical data is embedded within a statistical model through training. These are then capable of quick decision making over unseen data. The System 2 in Model 1 is the full-fledged CBR system that compares the present problem to similar cases, and reasons over them to solve it.

We use two switching strategies within Model 1, to decide when to fall back to CBR. These yield two versions of Model 1, as below.

Model 1A; Confidence-based Switching: The confidence-based switching strategy operationalizes what humans do with meta-cognition, where switching is performed on realizing that the problem is in a space where the intuition is judged to be shallow. Many machine learning models are capable of providing a *confidence* associated with their decision. For example, the confidence of Support Vector Machines (SVMs) decisions is directly related to how far the input is, from the learned decision boundary. Under this strategy, the switching from the ML/DL decision maker (System 1) to the full CBR (System 2) is done when the ML/DL decision maker has low confidence (e.g., confidence below a threshold). This is illustrated in Fig 2.

Model 1B; Oracle-based Switching: The human switching process is not always rational and judgement-based. The previously illustrated tension between the lure of fast decision making (powered by the availability heuristic) and meta-cognition is often inadvertently swayed towards the former by the desire to make fast decisions, lack of information to judge confidence and/or other unwillingness to invest System 2. In cases when such a System 1 decision goes wrong, the human received feedback from the environment on the quality of the decision. While there is no second chance in many scenarios (e.g., driving, job interviews etc.), repeat attempts are allowed in certain real-world cases (e.g., allowance of multiple attempts at a driving test, or certification of language fluency etc.). Model 1B, Fig 2, illustrates such a scenario where the system has access to an oracle or an expert to verify the correctness of the solution proposed by System 1. On receiving negative feedback, the model would trigger the more deliberate System 2. For example, a person solving a multiple-choice question with three options *a*, *b* and *c* may intuitively choose one option. However, on being given feedback that the chosen option is incorrect, he/she will begin solving the same problem more deliberately.

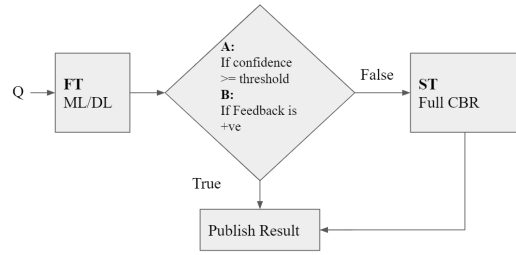


Figure 2: Model 1A and 1B Illustration; Q is the query or test input; Feedback is +ve implies System 1 solution is correct.

3.2 Model 2: Parsimonious CBR & Full CBR

The cognitive process of fast thinking, by virtue of being fast, does not have the luxury of making use of all the details of the decision making context. Here, we use this cue to model System 1 as a parsimonious CBR system that makes use of only a few key attributes. The System 2, as previously, remains the full-fledged CBR system which makes use of all attributes in decision-making. Given that it is infeasible to quantify the confidence of decisions that stem from the parsimonious nature of the System 1 CBR, we use an innovative tagging-based model to perform the switching. We also make use of feedback-based switching as in the case of Model 1. Additionally, we employ switching based on both tags and feedback as a hybrid switching model.

Model 2A; Tag-based Switching: Consider the set of all cases in the case base. We now undertake a per-case tagging process, whereby each case is tagged as either *FT* (fast thinking), *ST* (slow thinking) or *N* (none). This tagging is realized using a leave-one-out simulation of the decision-making process considering each case in turn. For each case *C*, we compute the *k* nearest neighbors (among the remaining cases in the case base) on the basis of the attributes used in the parsimonious CBR system (System 1/Fast Thinking). If *C* can be solved using those neighbors, we tag *C* as *FT*. In a classification setting, a case *C* is said to be *solved* when, *C*'s class label is same as the majority label among the *k* nearest neighbors. This is easy to check since all labels are available for the case base (akin to the training set in machine learning models). If the majority of labels of *C*'s System 1 neighbors do not match *C*'s label, we fall back to identifying *C*'s neighbors based on all attributes (System 2), and perform the solvability check as earlier. If this succeeds, *C* would be tagged as *ST* indicating it is solvable using the leave-one-out model using System 2 (ST). In case *C* is not solvable using either of the above, we tag it as *N*. Once this tagging process is completed in the pre-processing stage, we make use of it in the decision making stage. Our guiding heuristic is that the cases that fall within a neighborhood of *FT* cases are likely to be solvable using System 1/FT, whereas those that fall within a predominantly *ST* neighborhood would be better solved using System 2/ST. Accordingly, the working of Model 2A Fig 3, when it encounters a new case, is as follows:

- Identify the neighbors using the parsimonious CBR/System 1.

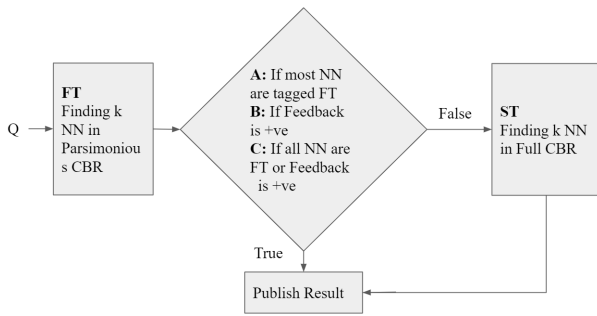


Figure 3: Model 2A, 2B and 2C Illustration.

- If the most abundant tag among them is *FT*, output the results based on System 1.
- Otherwise, find neighbors using full CBR/System 2, and output the results accordingly.

The threshold value used for making majority based switching can be varied based on the user’s response time and accuracy needs. If the threshold is too low, most of the queries will be solved by System 1, resulting in quick responses but low accuracy. In contrast, when the threshold is high, most of the queries will be handled by System 2, resulting in comparatively slower responses and better accuracy.

Model 2B; Oracle-based Switching: As in the case of Model 1B, 2B in Fig 3, makes use of oracle-based switching where feedback from an oracle or an expert, indicates whether the fast thinking solution is correct or not. If found incorrect, the reasoner will engage System 2.

Model 2C; Hybrid Switching: This hybrid switching model blends the tag and feedback based process. Whenever the System 1 neighbors are all tagged *FT*, we go ahead and report the System 1 decision. If not, we still invoke System 1 and use feedback to decide whether to switch to System 2. In other words, this is similar to the feedback model in 2B except that feedback is not sought when all neighbors of the test case bear the *FT* tag.

3.3 Model 3: Full CBR & CBR with Adaptation

In Model 3, we explore a digression from the earlier models, which is suited to scenarios where response time is not very critical. System 1 is modelled using full CBR (which was the System 2 in the previous models), whereas System 2 is modelled using a more elaborate process, called CBR adaptation. CBR adaptation, in addition to making use of the neighbors, uses the difference in the neighboring case and query case attributes in the solution process. This adaptation is often computationally demanding, thus making it more suited to be modelled within the System 2 processing. Similar to Models 1B and 2B, **Model 3B**, Fig 4, invokes System 2 when the expert feedback for System 1 response is ‘incorrect’. The other switching strategies like *Tag-based Switching* and *Hybrid Switching* in Model 2A and 2C respectively, can also be applied in Model 3, leading to **Model 3A** and **Model 3C**, Fig 4.

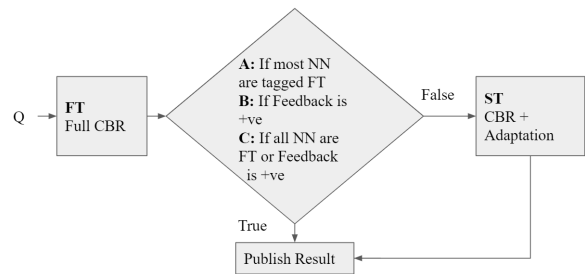


Figure 4: Model 3A, 3B and 3C Illustration.

4 Experimental details and Results

We have seen three models of realizing slow and fast decision making in CBR. In this section, we will discuss the experimental results for the same on different datasets. For Models 1 and 2, we have used the following with the classification datasets: *Pima Indians Diabetes*(PID) (Smith et al. 1988), *Fetal Health Classification* (FHC) (Ayres-de Campos et al. 2000), *Red Wine Quality* (RWQ) (Cortez et al. 2009) and *20 NewsGroup Dataset* (20NG) (Joachims 1996), having 768, 2126, 1599 and 2958 instances, respectively. Model 3 is evaluated over the *Graduate Admission Dataset 2* (GA2) (Acharya, Armaan, and Antony 2019) having 500 instances. The PID, FHC, RWQ, and 20NG datasets are having 2, 3, 2 and 3 class labels respectively. In 20 NG dataset only the documents of classes *sci.electronics*, *sci.med* and *sci.space* are used for classification. These labels form the solution, which is sought from each of the decision-making models. On each dataset, we have performed a 5-fold splitting where the data is spilt into five non-overlapping folds of the same size. We take one fold as the test or query dataset and the remaining four folds are used for training, that is, for building the casebase. This process is repeated by taking each of the five folds in turn for testing. The results are averaged over five runs and reported. All the experiments are done in Python 3.7 on a system with 8 GB RAM, Intel core i5 processor(8th gen) with Windows 10.

4.1 Experimental Setup

Model 1: In all our experiments related to Model 1, we have chosen Support Vector Machines (SVM) (Noble 2006) to represent the fast thinking model, which could, in principle, be any ML/DL model. System 2 is the basic CBR model, where the problem part of each case is having attributes and the solution part is the corresponding class label. We have empirically fixed the confidence threshold as 0.8 for Model 1A. In Model 1B, the feedback is binary: -1 and 1 for incorrect and correct respectively. When the confidence is less than 0.8 for model 1A or feedback is -1 for model 1B, the query is solved using CBR (System 2).

Model 2: In Model 2, both System 1 and System 2 are CBR models. In the fast thinking or parsimonious CBR, only a few attributes are chosen for representing the problem part whereas in the slow thinking or System 2 CBR, all attributes are used for computing problem side similarity. For all the datasets, System 1 attributes are chosen using the information gain. This may be replaced by domain expertise as and

when feasible. The PID dataset has 8 attributes and the following 3 were chosen for fast thinking: Glucose, BMI, and Age. In the FHC dataset, out of 21 attributes, the following 3 were chosen for System 1: Baseline Fetal Heart Rate, Value Accelerations, Abnormal Short Term Variability. Similarly, for RWQ dataset, important attributes were Sulphates and Alcohol out of 11 attributes in the dataset. In 20 NG the attributes are TF-IDF weights and top 200 features are selected for fast thinking, using information gain. In Model 2B the feedback value is the same as in Model 1B. In Model 2A, if the majority tag among the neighbours is FT, then the decision given by System 1 is used. Else, System 2 is invoked.

Model 3: Model 3 is evaluated on GA2 dataset where the query is initially solved using Full CBR (FT component). Based on external feedback, System 2 may be invoked for adaptation (ST component). System 2 uses locally weighted linear regression (LWLR) for adaptation and can be replaced by any model as appropriate. The time taken to solve a query is supposed to increase due to the extra computation involved in estimating the parameters of LWLR. The three switching strategies applicable for Model 2 are applicable for Model 3 as well.

4.2 Results and Analyses

Models 1 and 2: Table 1 shows the accuracies achieved by the proposed dichotomous models (1A, 1B, 2A, 2B, 2C) as well as two singular models (Full CBR and SVM) on the following four datasets: 20NG, RWQ, FHC and PID. Here, *Full CBR* refers to the normal CBR system where all attributes are used for similarity estimation. It is interesting to note that all dichotomous models, with the exception of Model 2A on the FHC dataset, perform better than the singular models. This drives home the inference that the cognitively inspired dichotomous models improve upon the singular ones. In particular, the dichotomous models are able to make the best of the following two worlds: model based generalization by fast thinking and individual experience based predictions by slow thinking. Further, we can also infer the following three trends from Table 1: *First*, Models 1A and 1B perform better than Models 2A and 2B, respectively, showing that blending ML and CBR models are more effective than blending parsimonious and full CBR models. *Second*, the feedback-based models (1B and 2B) perform better than their non-feedback-based counterparts (1A and 2A) which is as expected owing to their access to an oracle or expert. However, it is notable that the difference is not sizeable, further indicating that the switching strategy is able to stand-in for feedback fairly effectively. *Third*, the tag/feedback hybrid switching model in 2C, which does not consult the oracle as much as 2B, does, as expected, fares between the other two models in terms of accuracy.

Table 2 shows the response times averaged over all test queries for the proposed models. The significant difference in response times between FT and ST models shows that it would be useful to trigger switching only when absolutely necessary for efficiency considerations. Therefore, we emphasize the need for further research in developing good switching strategies. Further, we can also see that only a

Model	Switching Strategies	20NG	RWQ	FHC	PID
Model1 (ML/DL & Full CBR)	Confidence-based (1A)	92.66	92.39	90.64	81.30
	Oracle-based (1B)	93.78	94.91	91.25	85.06
Model2 (Parsimonious CBR & Full CBR)	Tag-based (2A)	88.46	86.56	77.84	75.97
	Oracle-based (2B)	95.66	92.13	90.82	84.80
	Hybrid (2C)	90.15	93.64	83.67	82.85
Full CBR		87.30	84.56	85.36	73.90
SVM		87.08	86.38	84.33	75.45

Table 1: Results for Accuracy (%).

Model	Switching Strategies	20NG		RWQ		FHC		PID	
		Time	ST%	Time	ST%	Time	ST%	Time	ST%
Model1 (ML/DL & Full CBR)	Confidence-based (1A)	1.56	13.82	169.23	16.00	717.8	26.58	88.98	46.23
	Oracle-based (1B)	1.53	12.98	137.15	13.63	430.75	15.67	49.51	24.55
Model2 (Parsimonious CBR & Full CBR)	Tag-based (2A)	1.69	4.68	430.86	13.25	686.33	9.89	156.58	22.86
	Oracle-based (2B)	1.83	12.22	451.34	17.32	1045.25	26.83	155.82	33.77
	Hybrid (2C)	1.75	4.06	434.8	12.19	844.9	14.92	159.09	22.73
Full CBR	-	3.75	-	1054.94	-	2759.59	-	191.49	-

Table 2: Results for average per query response time, Time (ms) and % query cases passed on to System 2, ST%.

small percentage of query cases are being passed on to ST in the dichotomous models. Hence, the increased accuracies of these models over singular models imply that in dichotomous models, CBR is effectively used i.e., it is used only for those exceptional cases which cannot be handled by the FT models. We have also studied the impact of thresholds on the tag-based switching model (Model 2A). In Fig 5, we can see that both accuracy and response time of the system (Model 2A) increases with an increase in threshold. As discussed in section 3.2 the user can fix the threshold value as needed.

Model 3: Table 3 shows the results of regression on GA2 dataset. In Model 3, the adaptation component is the LWLR model which is selectively triggered based on the switching mechanism. As expected, the dichotomous models, which are able to adapt their solutions, perform fairly better than Full CBR with no adaptation, though with a slight tradeoff in response time. Also, the variations in accuracy with change in switching strategy, again, emphasizes the importance of switching strategies in dichotomous models.

5 Discussion and Related Work

Let us examine a few more interesting directions of realizing the slow-fast dichotomy in CBR. As observed earlier, fast thinking is successful in relatively well aligned regions of the case base (Craw and Aamodt 2018), explained in Section 1. This also paves the way for learning better indices

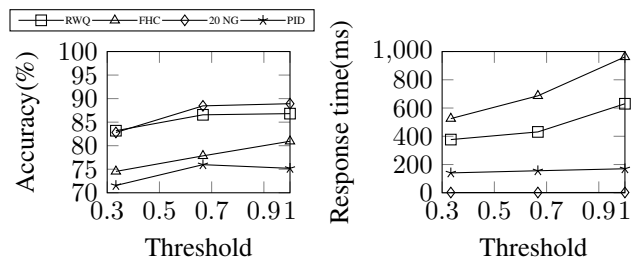


Figure 5: Model 2A, Accuracy and Response time (average per query) tradeoff, based on threshold.

Model	Switching Strategies	Accuracy (in%) (for 10% APE)	Average per query FT time (ms)	Average per query ST time (ms)	% Cases passed on to ST
Model 3 (Full CBR & CBR with adaptation)	Tag-based (3A)	67.6	132.22	311.93	25.4
	Oracle-based (3B)	78	121.14	311.93	33.6
	Hybrid (3C)	72.8	140.96	311.93	15.2
Full CBR	-	64.6	121.14	-	-

Table 3: Results for Model 3, on GA2 Dataset; APE: Acceptable Prediction Error (fixed by user).

over time – an area that has received significant attention in the CBR community. (Fox and Leake 1995) present an early effort where introspective reasoning processes are triggered by retrieval failures and used to refine indices in a CBR planner. There are also situations where slow thinking may entail a step of post-processing to verify the solution generated by fast thinking by checking whether the solution is consistent with constraints imposed by domain knowledge. Kahneman cites an interesting example, where subjects underestimate the crime rate in Michigan – while they are aware of Detroit as a high crime city, fast thinking fails to account for the fact that number of crime cases in Michigan should be inclusive of those in Detroit. In a recent work, the idea of holographic CBR (Ganesan and Chakraborti 2020) was presented, where cases have localized knowledge containers and thereby are informed of their association with the rest of the case-base. It was observed by the authors that update operations on such a richly connected network of cases are considerably slower than those in traditional casebases.

The need for realizing the slow-fast dichotomy has been realized outside the CBR context as well, though some of the past work do not explicitly refer to Kahneman’s work. Of particular interest is the class of anytime algorithms (Zilberstein 1996) that have the ability to give increasingly better results when allowed more computation time. (Grubb and Bagnell 2012) identify an interesting application in autonomous navigation of robots, explained in Section 1. (Dileep and Chakraborti 2016) identifies the need to flexibly switch between two modes of text classification, one that uses few easy-to-extract features and the other that is slower but more accurate since it relies on richer representations. (Peng et al. 2002) present a k-NN meta-classifier to choose between several classification algorithms – the idea of meta-learning is appealing in that it allows for interesting extensions to scenarios that demand anytime algorithms.

6 Conclusion

The paper presents diverse cognitively appealing realizations of systems that can flexibly trade off retrieval accuracy with respect to time efficiency by switching from fast to slow thinking based on either external feedback or introspection. In particular, we have seen that Machine Learning can complement CBR in realizing fast and slow reasoning processes respectively. A second approach is to realize fast thinking using retrieval over a subset of attributes; slow thinking entails a more exhaustive retrieval over the entire set of attributes, similarity estimation over some of

which could be computationally intensive. A third approach is where slow thinking uses adaptation to refine the results of fast thinking, thereby increasing effectiveness at the expense of speed. The empirical findings reported in the paper suggest that the approaches hold promise in meeting the demands of time constrained applications that need an efficient solution for most queries, and can only selectively afford a slow and more exhaustive retrieval in cases where fast retrieval fails. We envisage that the work will pave way for other interesting realizations not only in the context of CBR, but AI in general.

References

- Acharya, M. S.; Armaan, A.; and Antony, A. S. 2019. A comparison of regression models for prediction of graduate admissions. In *2019 ICCIDS*, 1–5. IEEE.
- Ayres-de Campos, D.; Bernardes, J.; Garrido, A.; Marques-de Sa, J.; and Pereira-Leite, L. 2000. Sisporto 2.0: a program for automated analysis of cardiocograms. *J Matern Fetal Neonatal Med* 9(5):311–318.
- Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decis. Support Syst.* 47(4):547–553.
- Craw, S., and Aamodt, A. 2018. Case based reasoning as a model for cognitive artificial intelligence. In *ICCB*, 62–77. Springer.
- Dileep, K., and Chakraborti, S. 2016. Eager to be lazy: Towards a complexity-guided textual case-based reasoning system. In *ICCB*, 77–92. Springer.
- Fox, S., and Leake, D. 1995. Using introspective reasoning to re ne indexing. In *IJCAI*, volume 391397.
- Ganesan, D., and Chakraborti, S. 2020. Holographic case-based reasoning. In *ICCB*, 144–159. Springer.
- Grubb, A., and Bagnell, D. 2012. Speedboost: Anytime prediction with uniform near-optimality. In *Artificial Intelligence and Statistics*, 458–466.
- Joachims, T. 1996. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.
- Kahneman, D. 2011. *Thinking, fast and slow*. Macmillan.
- Kolodner, J. 2014. *Case-based reasoning*. Morgan Kaufmann.
- Massie, S.; Craw, S.; and Wiratunga, N. 2006. Complexity profiling for informed case-base editing. In *ECCBR*, 325–339. Springer.
- Metcalfe, J.; Shimamura, A. P.; et al. 1994. *Metacognition: Knowing about knowing*. MIT press.
- Noble, W. S. 2006. What is a support vector machine? *Nature biotechnology* 24(12):1565–1567.
- Peng, Y.; Flach, P. A.; Soares, C.; and Brazdil, P. 2002. Improved dataset characterisation for meta-learning. In *DS*, 141–152. Springer.
- Smith, J. W.; Everhart, J.; Dickson, W.; Knowler, W.; and Johannes, R. 1988. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the ASCA in Medical Care*, 261.
- Zilberstein, S. 1996. Using anytime algorithms in intelligent systems. *AI magazine* 17(3):73–73.