

Learning and Exploiting Context in Agents

Bruce Edmonds

Centre for Policy Modelling,
Manchester Metropolitan University,
Aytoun Building, Aytoun Street, Manchester, M1 3GH, UK.
+44 161 2476479

<http://www.cpm.mmu.ac.uk/~bruce>

ABSTRACT

The use of context can considerably facilitate reasoning by restricting the beliefs reasoned upon to those relevant and providing extra information specific to the context. Despite the use and formalization of context being extensively studied both in AI and ML, context has not been much utilized in agents. This may be because many agents are only applied in a single context, and so these aspects are implicit in their design, or it may be that the need to explicitly encode information about various contexts is onerous. An algorithm to learn the appropriate context along with knowledge relevant to that context gets around these difficulties and opens the way for the exploitation of context in agent design. The algorithm is described and the agents compared with agents that learn and apply knowledge in a generic way within an artificial stock market. The potential for context as a principled manner of closely integrating crisp reasoning and fuzzy learning is discussed.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: *Intelligent agents*.
I.2.6 [Learning]: *Induction*. I.6.8 [Model Development]:
Modeling methodologies. I.5.3 [Clustering]: *Algorithms*.

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Context, integration, learning, deduction, genetic programming, evolutionary computation, cognitive analogy, biological analogy.

1. INTRODUCTION

In 1971 in his ACM Turing Award lecture, John McCarthy suggested that the explicit representation and manipulation of context might be a solution to the effective lack of generality in many AI systems (these ideas were later developed and written up in [21]). Since then context and context-like ideas have been investigated in both the AI and ML communities, culminating in several workshops [1-5] and a series of international conferences entirely devoted to the subject [8, 11]. However despite this attention, context-related techniques and ideas have not been explicitly applied to the design of autonomous agents to any significant extent.

Part of the reason for this may be the difficulty in entering (or otherwise specifying) the information relevant to each context. The main part of this paper aims to show a solution to this

problem by exhibiting a practical way in which agents can learn context-sensitive information about their environment. Another part of the reason may be that really exploiting context involves a close integration of learning and inference. Thus it straddles the AI and ML communities which are notoriously disjoint and mutually suspicious. In the last section I suggest that context can provide a well-motivated and coherent mechanism for the *close* integration of learning and deductive processes.

2. ABOUT CONTEXT

There are a great many different conceptions and uses of "context". In this section I briefly preview some of these in cognitive science, AI and ML, before proceeding to my analysis of the roots of context in Section 3. The word "context" is used both for the type of circumstance that allows for knowledge to be applied (sometimes called the 'external' context) as well as the cognitive structures that correspond to these (the 'cognitive' context). Since the design of agents focuses on the cognitive mechanisms of agents I will mean cognitive contexts, unless I say otherwise (see Edmonds for a discussion of the connection).

2.1 Context in Cognition

The use of context is a pervasive heuristic in human cognition. It appears that we use context in almost every area of our thinking and action, including: language understanding; memory; concepts and categorization; affect and social cognition and (probably) problem solving and reasoning [20]. In the past some researchers perceived the context-dependency of human thought purely as a disadvantage or side-effect, but now it is becoming increasingly clear that it is an *essential* tool for enabling effective learning, reasoning and communication in a complex world.

Although human cognition is not a necessary starting point for motivating the design of agents it is a fruitful one, especially when looking for solutions that will scale up to cope with problems of real world complexity.

2.2 Context In AI

McCarthy's idea was to reify the context to a set of terms, i , and the introduces an operator, ist , which basically asserts that a statement, p , holds in a context labelled by i . Thus:

$$c : ist(i, p)$$

read " p is true in context i " which is itself asserted in an outer context c . ist is similar to a modal operator but the context labels are terms of the language. Reasoning within a single context operates in a familiar way., thus we have

$$\forall i(ist(i, p) \wedge ist(i, p \rightarrow q) \rightarrow ist(i, q)).$$

In addition one needs to add a series of ‘lifting’ axioms, which specify the relation between truth in the different contexts. For example if $i \geq j$ means that “ i , is more general than context, j ”, then we can lift a fact to one of its supercontexts using:

$$\forall i \forall j (i \geq j) \wedge \text{ist}(p, i) \wedge \neg \text{ab}(i, j, p) \rightarrow \text{ist}(p, j)$$

where ab is an abnormality predicate for lifting to supercontexts.

This framework is written up in [22]. There are a whole series of formal systems which are closely related to the above structure, including, notably, Gabbay’s fibered semantics [15] and the local semantics of the *Mechanized Reasoning Group* at Trento [17]. A useful survey of such formalisms is [9].

Trying to apply generic reasoning methods to context-dependent propositions and models, will be either inefficient or inadequate [18]. The generic approach forces a choice of the appropriate level of detail to be included, so that it is likely that *either* much information that is irrelevant to the appropriate context will be included (making the deduction less efficient) *or* much useful information that is specific to the relevant context may be omitted (and hence some deductions will not be possible). The role context can play in solving the under/over determination of knowledge will be discussed in the last section.

2.3 Context In ML

The use of context in machine learning can be broadly categorized by goal, namely: to maintain learning when there is a hidden/unexpected change in context; to apply learning gained in one context to different context; and to utilise already known information about contexts to improve learning. There are only a few papers which touch on the problem of learning the appropriate contexts themselves. Widmer [30] applies a meta-learning process to a basic incremental learning neural net; the meta-algorithm adjusts the window over which the basic learning process works. Here it is an assumption that contexts are contiguous in time and so a time-window is a sufficient representation of context. Harries et al. [19] employ a batch learner as a meta-algorithm to identify stable contexts and their concepts; this makes the assumption that the contexts are contiguous in the “environmental variables” and can only be done off-line. Aha describes an incremental instance based-learning which uses a clustering algorithm to determine the weight of features and hence implicitly adjusts to context [6].

Other techniques require the explicit identification of what the contextual factors will be and then augment the existing machine learning strategy with a meta-level algorithm utilising this information (e.g. [27]). Others look to augment strategies using implicit information about the context to adjust features of the learning such as the weightings [6], or normalisation [26].

Turney discusses the problem in [28]. He surveys the various heuristics tried to mitigate the effects of context on machine learning techniques in [29]. He maintains a bibliography on context-sensitive learning at URL:

<http://extractor.iit.nrc.ca/bibliographies/context-sensitive.html>

2.4 Context in Natural Language

It has been recognized for a while that the external (and linguistic) context plays a role in the understanding of natural

language. However it is only recently that the importance of context in communication has been appreciated. The external context is not merely a resource for understanding utterances that is accessed when all other mechanisms fail; a way of sorting out otherwise ambiguous sentences. Rather it is one of the primary mechanisms. As Gardenfors [16] said:

Action is primary, pragmatics consists of the rules for linguistic actions, semantics is conventionalised pragmatics and syntax adds markers to help disambiguation (when context does not suffice).

In terms of developmental stages (and surely it is right to think of our agents as in the earliest stages of development) it is context that provides the meaning of specific parts of language. Thus natural language is rooted in context, allowing two individuals to guess at the contexts of others and hence share contexts. Such an ability to mutually identify the relevant context of communication lessens the need for formal and fixed ontologies.

3. THE ROOTS OF CONTEXT

In this section I recapitulate the analysis in [12] to motivate the learning algorithm to be presented. This argues that, causation is essentially a context-dependent abstraction. That in order to be able to effectively learn and reason about the world using fairly definite (i.e. ‘crisp’) models an agent has to separate out the foreground causes from the background ones (which can be abstracted to a context). This is illustrated in Figure 1.

The ‘background’ causes are those that are either so consistent that they can safely be ignored, or else are a messy mixture of factors capable of being recognized with a high probability afterwards but not explicitly incorporated into a reasonably simple “crisp” model. This will depend somewhat upon is usual in any particular circumstances. So, for example, if a man breaks a leg while walking down a step, the relevant foreground ‘cause’ would be his medical condition if he had brittle bone syndrome but due to his being distracted if a stripper ran by.

The model is thus learnt in one set of circumstances that are implicitly encoded by some recognition machinery (e.g. neural net). Later when the circumstances are recognized as being similar, the model is judged relevant to be included in any explicit reasoning or formal deduction. Thus knowledge is transferred from the time of learning to the time of application.

For such a transference to be possible a number of conditions need to be met, namely:

- that some of the possible factors influencing an outcome are separable in a practical way;
- that a useful distinction can be made between those factors that can be categorized as foreground features and the others;
- that the background factors are capable of being recognized later;
- that the world is regular enough for such models to be learnable;

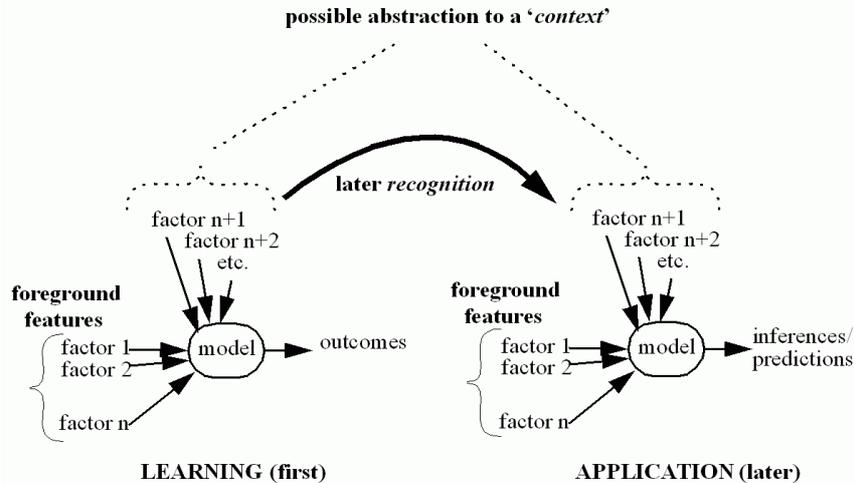


Figure 1. Reusing an earlier learnt model

- that the world is regular enough for such learnt models to be useful when applied in contexts that can be recognized later.

It should be clear that such a transfer of knowledge is not necessarily possible, because it relies on the presence of commonalities in the domain that is being interacted with. Broadly these commonalities must be fairly constant during the learning and application events (otherwise they might not be background), and be recognisable from one to the other. Different commonalities result in different sorts of context. For example: two agents might be inhabiting a common location in space and time and hence can use that as the context for communicative acts; or one may remember what it is like during a stock market crash previously and hence have some ready made models of how to act during another one.

While this transference of learnt models to applicable situations is the basic process, analysts of this process might abstract some aspects of the background features as a 'context'.

Note that the agent might not be able to explicitly identify and label the contexts that it is using, even if this is clear to an exterior observer. All that is necessary is for the agent to recognise the circumstances where models can be applied, or at least find the 'closest' candidate models in terms of their domain of application. On the other hand the agent *might* be able to introspect sufficiently to analyse and abstract its own contexts. It would seem that we, as humans, are so good at automatically flipping between different cognitive contexts that we do not notice this most of the time, but simply deal with reasoning within the chosen context. There are exceptions of course; for instance when trying to generalise to a theory or when trying to find out what went wrong.

Given the above conditions are possible context is:

an abstraction of those background elements of the circumstances in which a model is learnt that allows the recognition of new circumstances where the model can be usefully applied.

Due to the fact that context is characterised as an abstraction of an aspect of a heuristic for the learning and application of

knowledge, the properties of such contexts can not be meaningfully analysed if one only considers either the learning or the application of such knowledge. If one did this one would not only be missing out on over half of the story but also undercutting the reasons for its very existence. If the problems of learning are ignored then there is no reason not to encode such models without context – the non-causal factors can be treated as either given or the same as the other features of the model, de-contextualising them. If the problems of inference are ignored then there is no reason to separate the recognition of an appropriate context from that of recognising the correct prediction in that context. Thus if one is to exploit the power of context, both learning and inference need to be included.

4. LEARNING CONTEXT WITH CONTENT

In order for context-dependent reasoning to occur, the context-dependent information (or beliefs) need to be captured. If the relevant contexts are already known by the designer (and there is some effective way of recognizing when they apply), then either the relevant information can be entered or a context-enhanced learning algorithm can be employed to learn the information with respect to each context. The former case can be onerous because one not only has to enter the relevant facts as well as specifying each fact's domain of application, but one also has to define all the 'lifting-rules' to allow the integration of the context-dependent information. In the later case the context-dependency of the learning means that one needs correspondingly more information within each context for the learning to be complete.

Thus in order for the desired efficiency in terms of context-constrained reasoning to occur (without a laborious entry of information) for each appropriate context, this information (that is both the contexts and the content in the contexts) should be learned by the agent, at least to some extent.

4.1 The Context Learning Algorithm

The basic idea is to simultaneously learn the models and the circumstances in which they work best. If there is sufficient regularity in the environment to allow it this will allow some

clusters of similar circumstances to be identified and the corresponding models to be induced. However the clustering and induction parts of the algorithm can not work independently; i.e. clusters of like circumstances being identified and then models induced for these clusters. The reason for this is the contexts are identified by those circumstances where particular models work best. These may correspond to a neat (i.e. humanly identifiable) cluster but this is not inevitable – they may be (to the human eye) inextricably intertwined or overlapping.

There is a population of candidate beliefs, each of which is composed of two parts: a crisp model in a formal language (the *content*) and some information that specifies the model's domain of application (the *domain*). In the examples given here the designer specifies what inputs will be used for context recognition and which can be referred to in the model content (some may be in both). Repeatedly a particular circumstance is chosen (for example, these are the ones that simply occur to the agent), and those beliefs who are recognized as most probably relevant (or 'closer') are selected. Out of these the ones that work best are preferentially selected and crossed into future generations of the population. Beliefs that are never anywhere near occurring circumstances are, over time, forgotten.

The basic learning algorithm is as follows:

Randomly generate candidate models and place them randomly about the domain, D

for each generation

repeat

randomly pick a point in D, P

pick n models, C, biased towards those near P

evaluate all in C over a neighbourhood of P

pick random number x from [0,1)

if x < propagation probability

then propagate the fittest in C to new generation

else cross two fittest in C, put result into new generation

until new population is complete

next generation

A biological analogy makes this clear. Imagine that each belief is an plant. These plants exist in a space defined by the factors that allow context recognition. They compete locally, and those that are better replicate themselves into a neighbourhood (by propagation and sexual reproduction). Thus slowly the successful plants adapt and spread to fill all of the space in which they are relatively successful. Different plants will occupy different areas in the space. The contexts correspond to the ecological niches.

This is an example of the some more general heuristics for learning context.

Formation: A cluster of models with similar or closely related domains suggests these domains can be meaningfully abstracted to a context.

Abstraction: If two (or more) contexts share a lot of models with the same domain, they may be abstracted (with those shared models) to another context. In other words, by dropping a few models from each allows the creation of a super-context with a wider domain of application.

Specialisation: If making the domain of a context much more specific allows the inclusion of many more models (and hence useful inferences) create a sub-context.

Content Correction: If one (or only a few) models in the same context are in error whilst the others are still correct, then these models should either be removed from this context or their contents altered so that they give correct outputs (dependent on the extent of modifications needed to "correct" them)

Content Addition: If a model has the same domain as an existing context, then add it to that context.

Context Restriction: If all (or most) the models in a context seem to be simultaneously in error, then the context needs to be restricted to exclude the conditions under which the errors occurred.

Context Expansion: If all (or most) of the models in a context seem to work under some new conditions, then expands the context to include these conditions.

Context Removal: If a context has only a few models left (due to principle 2) or its domain is null (i.e. it is not applicable) forget that context.

These, the above algorithm and its properties is discussed in much greater detail in [13].

4.2 Example: Agents In An Artificial Stock Market

In order to demonstrate this approach to learning, I needed an environment that was sufficiently complex yet having emergent contexts (i.e. ones difficult to predict in advance). I have chosen a stock market model, composed of many trading agents and one market maker (roughly following the form and structure of [24]). The traders can choose to buy or sell one of a number of shares (if this is possible for them) from or to the market maker. The only fundamental in the market is a dividend rate for each of the shares which slowly change in a random walk. There are only a limited amount of each stock available to the market as a whole. The market maker sets prices as a result of the demand - if there is net demand for a stock it raises the price and if there is a net negative demand it lowers the price. There is a small transaction cost to the traders for every trader, so rapid random trading is unlikely to benefit it.

The goal of the traders is to maximise the total value of their assets (cash plus shares at current value). Thus the traders are in competition with each other – one trader tends to gain at another's expense. However this is not a zero-sum game due to the dividends paid on stocks and the possibility of making money at the market maker's expense.

Each time period the traders simultaneously buy or sell each of the stocks, assuming they have enough cash to fund the net price, the stocks to sell, and the market maker has the stocks to sell. Traders do not have to trade in any stock. Thus the decision that each of the traders has to make is how much to attempt to buy or sell of each stock each time period.

Traders can observe the following:

- the current and past prices of all stocks;
- the past actions of all traders;

- the current and past dividend rates.

In addition the traders are provided with primitives for:

- the current and past market index (average of all prices);
- recent trend of the index;
- recent total volume of trading;
- recent market volatility;
- the maximum historical price of any stock.

The operators available to the agents to build models with are:

- basic arithmetic (+, -, ×, ÷);
- the ability to refer back in time (last and lag operators).

They also have some constants, namely:

- the names of the other traders,
- the names of the stocks
- and a selection of random constants.

Basically the traders try to learn to predict what each of the stocks will be in the next time period and then buy or sell if they predict it will rise or fall sufficiently for this to be worthwhile.

This sort of set-up produces a rich series of dynamics as the traders participate in sequences of modelling ‘arms-races’ and imitation ‘games’. Any successful prediction schema will not last forever as the other traders will soon spot your trading pattern and exploit it to your disadvantage. However, as with real stock markets, there are definitely patterns and market ‘moods’ (if there are enough traders and stocks), for example bull markets and speculative bubbles. There will be periods of relative quiet as traders sit on stock and so effectively prevent trading and periods of high volatility as subgroups of traders engage in bouts of activity trying to exploit each other. The dynamics are related to those of the “minority game” [10], and similar [7] but are more varied and complex. Thus, although this is an artificial setting, it goes way beyond a “toy” problem in scope and complexity.

There are two types of traders: which I will call *generic* and *context* traders. Both types maintain a population of 20 models, each of which is composed of a separate expression to predict the

future price of each stock. All models are initially randomly generated to a depth of 5 using the inputs, primitives, operators and constants already listed. Both agents use an evolutionary learning algorithm which evaluates fitness by the profit the agent would have made over the past 3 time periods had it used these models to predict prices.

The generic traders use a genetic programming learning algorithm to evolve their predictive models and the context traders have an adapted version of this algorithm to allow the simultaneous learning of context for its models. The types are otherwise identical.

The learning algorithm for the generic trading agent is as follows:

```

Randomly generate initial population of candidate models
for each generation
  for each model
    evaluate what the total wealth of the agent would be if
    it had used this model in trading over the past few
    time periods, this is the model's fitness
  next model
  repeat
    randomly pick two models with a probability proportional
    to their current fitnesses
    pick random number x from [0,1)
    if x < propagation probability
      then propagate them to new generation
      else cross them and put results into new generation
    until new population is complete
  next generation

```

The context trader’s algorithm differs a little from the basic version outlined in the last section. This is because from an agent’s point of view the only relevant circumstances (in terms of the space of possible ones) are those that *actually* occur. Therefore instead of randomly picking a sequence of circumstances until the new population is generated, we use only the present circumstance repeatedly and we propagate the rest into the next population with a bias against those that are furthest from any circumstance that has occurred. Also in this model we have associated with each model content a set of positions, so that its *domain* of application is indicated by a small

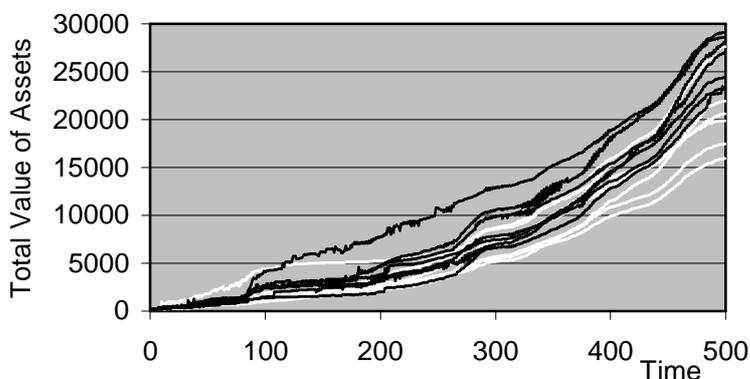


Figure 2. Growth in Agents’ Assets over time (context traders in black, generic in white)

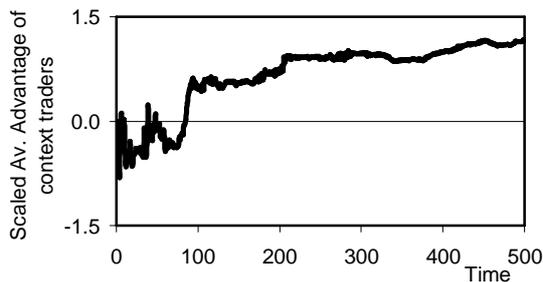


Figure 3. Difference of average asset values of context and generic traders, scaled by current asset spread

cloud of points.

It is not obvious that the context trader is a better learner than the generic trader. The context algorithm restricts which models can be crossed to produce new variants to those that are in the same neighbourhood of an occurring circumstance, whilst the generic algorithm allows a more global search for solutions. Thus one might expect that the context traders do better only if there is a context-dependency in the environment to exploit. As we shall see this appears to be the case in this model.

The model was run with 7 of each type of agent (thus 15 including the market maker) trading 5 different stocks over 500 time periods. The model was implemented in SDML [23].

For the first 80 periods one of the generic traders was doing substantially better than the others, but after this the context traders clearly did better, on the whole (see Figure 2). To make clear the significance of the difference between context and generic traders I have plotted the difference between the average value of context traders' assets minus the average value of the generic trader's assets, scaled by the current standard deviation of the spread of total asset values (Figure 3).

It is notable that the generic traders did better if there were only 2 or 3 of each type of trader – the context traders only reliably out-perform the generic traders (on the whole) with larger populations of traders. The context traders do particularly well if they are in a minority among many generic traders. It is

postulated that it is only with larger numbers of the same type of trader that learnable contexts appear in the trading patterns for the context traders to learn and exploit.

To show that the context traders are, in fact, identifying meaningful contexts (at least sometimes), I have taken a snapshot of the positions indicating the domain of the 6 of the models in one agent for one stock at one time (the best performing agent halfway through the run). These clusters are shown in Figure 4. The contents of these six model are shown in Table 1.

Table 1: The action models (for stock 3) in Figure 3.

model-256	priceLastWeek [stock-4]
model-274	priceLastWeek [stock-5]
model-271	doneByLast [normTrader-5] [stock-4]
model-273	lDidLastTime [stock-2]
model-276	lDidLastTime [stock-5]
model-399	minus [divide [priceLastWeek [stock-2]] [priceLastWeek [stock-5]] [times [priceLastWeek [stock-4]] [priceNow [stock-5]]]

For this agent at this time there seem to be three contexts: one for lower volatility and higher volume, one for lower volatility and lower volume and one for higher volatility and middle volume. It is notable that, even within each of these there are a mixture of two models that are appropriate. Thus, even given the circumstances, the model selected for will be determined by recent predictive performance: for example, in the case of stock 3 in the above snapshot its price may be modelled best by either the price of stock 4 or stock 5 last time period.

5. DISCUSSION: PROSPECTS FOR THE INTEGRATION OF LEARNING AND DEDUCTION

Figure 4 and Table 1 above show the way context can separate

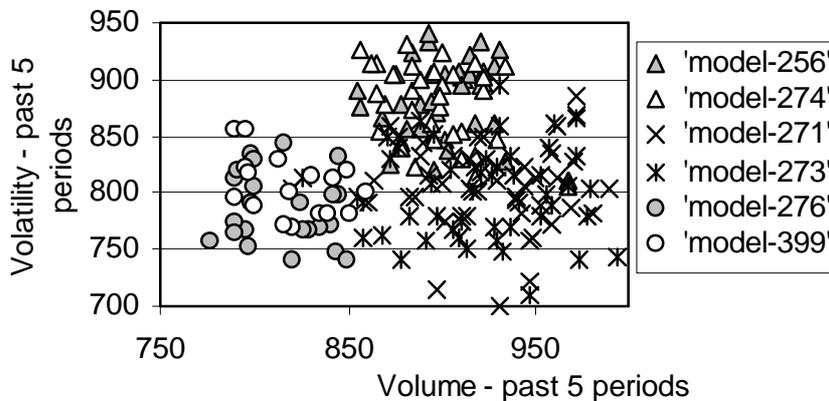


Figure 4. Snapshot of clusters of positions of 6 action models for a context trader indicating three distinct contexts.

the necessary ‘fuzziness’ of relevance decisions from the ‘crisp’ content models upon which deductive and planning algorithms could be usefully employed. The crispness of the content, N , is made possible by the restriction of its applicability to a recognisable context. If the domain was capturable in a crisp way, to a symbolic representation, X , the knowledge could be decontextualised: $X @ N$, but the point is that the domain is often not suitable to any compact symbolic representation but is a messy mixture of heuristic indicators. In fact there is a good argument to say that it is only feasible to reason about the complex natural world within fuzzily defined but restricted contexts. If the content, N , was of a similar nature to X then there would be no need for it because it could be subsumed into the recognition process. Thus the utility of context-dependency derives from its two aspects, it loses much of its point if reduced to either just the symbolic or non-symbolic aspects. Thus it straddles the ML and AI communities.

To illustrate the power of context-dependency, I will outline how it could be employed to solve some classic problems in AI, namely the *under-* and *over-*determination of knowledge. If an agent has a set of beliefs, B and is trying to decide whether to take a specific action, dependent on whether a predicate α is true or not, there are two problematic cases for it:

- (1) when neither α nor $\neg\alpha$ can be proved (under-determination);
- (2) when a contradiction is obtained, i.e. both β and $\neg\beta$ can be proved (destructive over-determination).

In (1) there is not enough knowledge to specify whether α nor $\neg\alpha$ is true. If the agent has a store of context dependent knowledge, it can then search for a more specific context, which may provide it with the extra information it requires.

In case (2), something is wrong with the agent’s set of beliefs. There are two possibilities: *firstly* that the agents has chosen the wrong context and *secondly* that there is something wrong with the beliefs associated with that context. Distinguishing between these possibilities is done by checking other consequences of beliefs within that context; if other predictions relevant to that context are also false then it is likely that the context has been wrongly recognised, in which case it is sensible to search for another (probably more general) context that might be appropriate; if the other predictions in the context are correct then it is likely that some of the specific beliefs used to infer b and $\neg b$ need updating or rejecting from this context.

Many non-monotonic logics can be seen as attempts to solve the above problems in a generic way, i.e. without reference to any contingent properties obtained from the particular contexts they are applied in. So, for example, some use ‘entrenchment’ to determine which extra information can be employed (e.g. oldest information is more reliable [14]), and others allow a variety of default information to be used (e.g. using extra negative knowledge as long as it is consistent [25]). These may work well on occasion and tolerably well in others, but the only truly reliable way to update knowledge in a context is by utilising the specific properties of that context. Combining the learning and deductive exploitation of context-dependent information should enable the effective and correct integration of learning and deduction.

Thus the introduction of context into the agent architecture would allow us to progress beyond the ‘loose’ loop of:

```
repeat
  learn and/up update beliefs
  deduce intentions, plans and actions
until finished
```

to a more integrated loop:

```
repeat
  repeat
    recognise/learn/choose context
    induce/update beliefs in that context
    deduce predictions/conclusions in that context
  until predictions are consistent
    and actions/plans can be determined
  plan & act
until finished
```

Only the recognition of a context and the final stage (plan & act) do not occur within the confines of a context. The recognition machinery can be parallel to the rest so that it is ready to suggest a context when called upon to do so.

6. Conclusion

Context has a huge potential for improving the performance of agents in multifaceted and unpredictable domains. It combines symbolic and non-symbolic forms of knowledge. It can make reasoning more efficient by structuring the space of knowledge by relevance. It allows the close and coherent integration of learning and deduction. It provides a partial solution to the problems of the under- and over-determination of knowledge and it holds out the potential for better and more flexible communication via the possibility of mutually identifying the relevant communicative context.

It is essentially a trade-off: more information is stored including the relevance information implicit in the contexts, so that more effectiveness can be obtained. It can be seen as a sort of pre-compilation of knowledge. It *hugely* increases the amount of information that needs to be stored. However, in the case of agents who are learning about the environment *in situ* this is merely a case of encoding and remembering the contextual information that is already available to them. What was missing was an effective way of capturing this contextual information. Algorithms similar to that presented here might provide this missing piece.

7. REFERENCES

- [1] 2nd European Conference on Cognitive Science, Workshop on Context. <http://www.cs.man.ac.uk/ai/ECCS97/>.
- [2] 1993 IJCAI Workshop on Using Knowledge in Its Context. <http://context.umcs.maine.edu/IJCAI93/>.
- [3] AAAI-95 Fall Symposium on Formalizing Context. <http://www-formal.Stanford.EDU/buvac/95-context-symposium/>.
- [4] IJCAI-95 Workshop on: Context in Natural Language Processing. <http://www.cs.wayne.edu/lucja/context-w1.html>.

- [5] AAAI99 Workshop on Reasoning in Context for AI Applications. <http://context.umcs.maine.edu/AAAI99-Workshop/>.
- [6] Aha, D.W., Incremental, instance-based learning of independent and graded concept descriptions. in 6th Int. Workshop on Machine Learning, (1989), Morgan Kaufmann, 387--391.
- [7] Akiyama, E. and Kaneko, K. Evolution of Cooperation, Differentiation, Complexity, and Diversity in an Iterated Three-person Game. *Artificial Life*, 2. 293-304.
- [8] Akman, V., Bouquet, P., Thomason, R. and Young, R.A. (eds.). Modeling and Using Context: Proceedings of the Third International and Interdisciplinary Conference, CONTEXT2001, Dundee, Scotland, 2001. Springer-Verlag, Berlin, 2001.
- [9] Akman, V. and Surav, M. Steps Toward Formalizing Context. *AI Magazine*, 17. 55-72.
- [10] Arthur, B. Inductive Reasoning and Bounded Rationality. *American Economic Association Papers*, 84. 406-411.
- [11] Bouquet, P., Serafini, L., Brézillon, P., M. Benerecetti and Castellani, F. (eds.). Modeling and Using Context: Proceedings of the Second International and Interdisciplinary Conference, CONTEXT'99, Trento, Italy, September 1999. Springer-Verlag, Berlin, 1999.
- [12] Edmonds, B. The Pragmatic Roots of Context. in Bouquet, P., Serafini, L., Brézillon, P., Benerecetti, M. and Castellani, F. eds. Modeling and Using Contexts: Proceedings of the Second International and Interdisciplinary Conference, CONTEXT'99, Springer-Verlag, Berlin, 1999, 119-134.
- [13] Edmonds, B. Learning Appropriate Contexts. in Akman, V., Bouquet, P., Thomason, R. and Young, R.A. eds. Modeling and Using Context, Springer-Verlag, 2001, 143--155.
- [14] Gärdenfors, P. Epistemic Importance and Minimal Changes of Belief. *Australasian Journal of Philosophy*, 62 (2). 136--157.
- [15] Gabbay, D.M. *Fibring logics*. Clarendon, Oxford, 1999.
- [16] Gärdenfors, P., The pragmatic role of modality in natural language. in 20th Wittgenstein Symposium, (Kirchberg am Weshel, Lower Austria, 1997), Wittgenstein Society.
- [17] Ghidini, C. and Giunchiglia, F. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127 (3). 221-259.
- [18] Greiner, R., Darken, C. and Santoso, N.I. Efficient reasoning. *ACM Computing Surveys*, 33 (1). 1-30.
- [19] Harries, M.B., Sammut, C. and Horn, K. Extracting Hidden Contexts. *Machine Learning*, 32. 101-112.
- [20] Kokinov, B. and Grinberg, M. Simulating Context Effects in Problem Solving with AMBR. in Akman, V., Bouquet, P., Thomason, R. and Young, R.A. eds. Modeling and Using Context, Springer-Verlag, 2001, 221-234.
- [21] McCarthy, J. Generality in Artificial-Intelligence - Turing Award Lecture. *Communications of the Acm*, 30 (12). 1030-1035.
- [22] McCarthy, J. and Buvac, S. Formalizing Context (Expanded Notes). in Westerstaahl, A.A.a.R.v.G.a.D. ed. *Computing Natural Language*. CSLI Publications, Stanford, California, 1998, 13--50.
- [23] Moss, S., Gaylard, H., Wallis, S. and Edmonds, B. SDML: A Multi-Agent Language for Organizational Modelling. *Computational and Mathematical Organization Theory*, 4 (1). 43-69.
- [24] Palmer, R.G.e.a. Artificial Economic Life - A simple model of a stockmarket. *Physica D*, 75. 264-274.
- [25] Reiter, R. A Logic for Default Reasoning. *Artif Intell*, 13. 81-132.
- [26] Turney, P. and Halasz, M. Contextual Normalization Applied to Aircraft Gas-Turbine Engine Diagnosis. *Applied Intelligence*, 3 (2). 109-129.
- [27] Turney, P.D., Robust classification with context-sensitive features. in *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE-93*, (Edinburgh, 1993), Gordon and Breach, 268-276.
- [28] Turney, P.D., The identification of context-sensitive features: A formal definition of context for concept learning. in *ICML-96 Workshop on Learning in Context-Sensitive Domains*, (Bari, Italy, 1996), 53-59.
- [29] Turney, P.D., The management of context-sensitive features: A review of strategies. in *ICML-96 Workshop on Learning in Context-Sensitive Domains*, (Bari, Italy, 1996), 60-66.
- [30] Widmer, G. Tracking Context Changes through Meta-Learning. *Machine Learning*, 27. 259-286.