Agent and Multiagent Systems Research Methodologies

Michael Rovatsos
School of Informatics
The University of Edinburgh
mrovatso@inf.ed.ac.uk

EASSS 2013 Tutorial
1st July 2011

Introduction

Objectives

- Get overview of agents research: history, areas, trends
- Revisit and critically review methodological foundations
- Provide insight into predominant approaches
- Evaluate and situate agents research contributions
- Broaden perspective to look beyond own specific work

Structure

This tutorial consists of four parts:

1. What is agents research?
   - Definitions, history, topics
2. How is agents research done?
   - Questions, methods, tools
3. How do we evaluate agents research?
   - Paradigms, criteria, presentation
4. How should we do agents research?
   - Methodological choices, practical guidance

About the speaker

- Active in agents research since 1999 ("2nd generation")
- Teaching agents and multiagent systems since 2004
- Work in multiagent learning, agent communication, social reasoning, trust, multiagent planning
- Experience with knowledge-based, numerical, declarative & game-theoretic methods
- Long-standing side-interest: methodologies in agents research

A note of caution

- Some material covers general research methodology issues that apply to any scientific discipline
  - Attempt to map it as concretely as possible to agents research
- Many views expressed here are strictly personal
  - Take them just as one opinion, and argue against it!
- Criticising research is a sensitive matter
  - I will try to keep criticism constructive and attempt presenting a balanced view

A note of caution

- Injustice cannot be avoided
  - Not everything that is important can be covered, but topics covered are important
- Some areas are underrepresented, material has a strong AI, theory & engineering bias
  - (social simulation, virtual agents, robotics, agent-oriented software engineering)
- The best way to deal with these issues: **Discuss!**

What is agents research?

Part I: What is agents research?

Opinions about agents research

A question worth asking?

- Game-theory & agents friend:

  I don't care what you call it. In fact, I even find "computer science" an overly narrow term. I go to economics conferences, theory conferences etc. as long as people have similar interests it's fine.

- Random AAMAS attendee:

  For me agents research is everything that gets published at AAMAS. If the community accepts it, it means it's agents research.

Opinions about agents research

How broad should the field be?

- Machine learning guy:

  I submitted something to AAMAS and it got accepted, although I knew nothing about the field - you guys are so much more open-minded than my community, it would be impossible to do that at NIPS.

- AAMAS attendee to colleague:

  Where are the agents in your work? It seems like you are presenting a centralised, collaborative system.

Opinions about agents research

What has the field achieved? What distinguishes it?

- Theoretical computer science professor:

  I have been hearing a lot about agents for 15 years now. Can you give me examples of key results that the field has produced so far?

- Senior AI colleague:

  I read those agents papers, and you guys use so many complicated words. I don't know what these words mean. What are meaningful, well-defined problems in multiagent systems that are different from other problems in AI?

Opinions about agents research

Who are our "customers"? What do we have to offer?

- Semantic Web veteran:

  When we developed all our standards, we thought you guys would supply the agents that use the information on the Web to do intelligent things. Why are there not many of them out there?

- Computer networking PhD student:

  I want to do adaptive routing for wireless devices, sounds like multiagent reinforcement learning would be an appropriate method. My devices need to make decisions within 10ms, can I use one of your algorithms?

The history of agents

- The Turing Test (1950)
  - Implies embodiment and interaction with an "agent"
- The Dartmouth conference (1956)
  - The dawn of "Artificial Intelligence"
- A universal modular actor formalism for artificial intelligence (1973)
  - First "agent" model of an autonomous interacting entity
- The Contract-Net Protocol (1980)
  - A first concrete procedure for task allocation
- Deals Among Rational Agents (1984)
  - Self-interested agents appear

The history of agents

- The use of meta-level control for coordination in a distributed problem solving network(1983)
  - Distributed problem solving, blackboard systems
- A robust layered control system for a mobile robot (1986)
  - Subsumption architecture – “Elephants don’t play chess”
- Plans and resource-bounded rational reasoning (1988)
  - The Belief-Desire-Intention model of agency
- Intention is choice with commitment (1990)
  - Formal foundations for deliberative architectures
- An abstract architecture for rational agents (1992)
  - Practical reasoning system architecture

The history of agents

- Communicative actions for artificial agents (1995)
  - Formal foundations for agent communication
- On the acceptability of arguments (1995)
  - Abstract framework for agents that argue
- Provably optimal boundedly-rational agents (1995)
  - Decision-theoretic AI and meta-reasoning
- Collaborative plans for complex group action (1996)
  - Foundations of teamwork and joint planning
- The Distributed Constraint Satisfaction Problem (1998)
  - Solving CSPs with multiple agents

The history of agents

- Shopbot economics (1999)
  - Agents and electronic commerce
- Algorithm for optimal winner determination in combinatorial auctions (2002)
  - Generalising resource allocation problems
- When are elections with few candidates hard to manipulate? (2007)
  - Complexity analysis of social choice problems
- The complexity of computing a Nash equilibrium (2009)
  - How hard is equilibrium computation in general?

Technology context

This research has been pursued against the backdrop of . . .

- the emergence of distributed computing
- the development of high-level programming languages
- enormous advances in the performance of computers
- radical shifts toward automation in all parts of society
- the evolution of the Internet and other global networks
- the increasingly ubiquitous presence of computers

Important to remember: most of agents research would be irrelevant without these

What is agents research?

Agents research = research about agents

- What are agents?
  - Autonomous, situated, persistent entities that interact with the environment through “sensors” and “effectors”
- What are computational agents?
  - Those implemented on computer software or hardware
- What are intelligent agents?
  - Reactive, proactive, rational, socially able agents
- As the area grew out of AI, there is a strong bias toward associating “agency” with “intelligence”
  - Areas like virtual agents, social simulation, agent-oriented software engineering emphasise other aspects

Pitfalls of the ”agents” concept

- The concept is excessively broad, under-specifies class of systems in question
  - What distinguishes them really from other systems?
  - Thermostat argument, vagueness of “autonomy”
- Anthropomorphism
  - That’s OK when this is the research objective
- Scope of functionality
  - Does an agent have to be able to do “everything”?
- How important is the “multi” according to this view?
  - Does “agent” really say anything about “multiagent”?

Pitfalls of the "agents" concept

- Debates sometimes not very fruitful
  - There seems to be a vague implicit concensus at least within a community
- Abuse as purely conceptual notion
  - Use of the metaphor for traditional systems ("agentification")
  - Use of agent-based methods (e.g. programming languages) for conventional systems
- Association with quality of solution
  - Concept suggests "comprehensive", not partial, solutions, and may sound better than "program"
- Ambiguity of agent concept might be a good thing
  - Enables mix of methods and exchange among sub-communities

The “holistic position”

Agents research is about integration of system components in a single system . . .

- Integrating an system in its environment
  - Autonomous, situated and embodied agents
- Integrating a system with its user
  - Personal assistants
- Integrating various capabilities in one system
  - Agent architectures

The "holistic position"

. . . and about integration and interaction among separate system components

- Integrating systems with shared goals in the same environment
  - Cooperative and benevolent agents
- Integrating several systems with different goals in the same environment
  - Self-interested agents – really novel aspect of computation
- Agents research = the science of integration?
  - Maybe, in its infancy. Another reason for advocating breadth?

Agents research topics

- Agent architectures
  - How to structure the design of an agent
- Agent reasoning & learning mechanisms
  - What algorithms to run on a single agent
- Agent interaction and communication mechanisms
  - How to make agents talk to each other
- Agent-based simulation of humans
  - How to build lifelike/believable agents
- Agent programming languages
  - How to implement agents

Agents research topics

- Multiagent coordination methods
  - How to make agents consider others
- Multiagent collaboration techniques
  - How to make agents work together
- Multiagent reasoning mechanisms
  - How to make joint decisions and reach agreement
- Multiagent-based social simulation
  - How to reproduce the behaviour of human collectives
- Multiagent learning techniques
  - How to learn in the presence of other learners

Methods used

- Social science, psychology, philosophy, linguistics
- Knowledge representation and reasoning
- Search, planning, and constraint satisfaction
- Formal methods, logic and automated verification
- Software engineering, human-computer interaction

Methods used

- Complexity and intractability analysis
- Game theory, economics, social choice theory
- Machine learning and pattern recognition
- Probabilistic reasoning, information & decision theory
- Efficient algorithms and combinatorial optimisation

Relationships to other fields of computing

- Decentralisation & interaction
  - Distributed systems, service-oriented computing, P2P etc
- Embodiment & autonomy
  - Robotics, sensor networks, mobile & pervasive computing
- Conflict of interest
  - Algorithmic game theory, eCommerce, game-playing AI
- Integration
  - Robotics, ”general AI”, collective intelligence, games
- Learning, emergence & self-organisation
  - Machine learning, evolutionary computation & swarms
- Use of methods
  - Theoretical computer science, numerical simulation, programming languages, human-computer interaction

Relationships to other disciplines

- Decentralisation & interaction
  - Sociology, social psychology, biology, linguistics
- Embodiment & autonomy
  - Electronics, mechanical & control engineering, design sciences, architecture
- Conflict of interest
  - Economics, politics, biology
- Integration
  - Design sciences, engineering, management science
- Learning, emergence & self-organisation
  - Biology, psychology, sociology, cybernetics
- Use of methods
  - Mathematics, philosophy, economics, biology

How is agents research done?

Part II: How is agents research done?

Introduction

- In this part we'll discuss the main strands of research in the current landscape
- Structured by classes of problems addressed by different communities
- For each of these classes, introduce and compare main schools of thought
- Comment on evolution of each community and current state of the art

Modelling agent-based systems

- Logic-based specification
  - Formal and computational logic: esp. epistemic logics and strategy logics
  - Declarative systems: close relation to logic programming, formal methods
- Mathematical specification methods
  - Probabilistic models of action and change: Markov Decision Problems etc.
  - Game-theoretic structures: normal-/extensive-form games, coalitional games, social choice problems
- Overlap
  - Logical languages and succinct representations for games and preferences

Single-agent reasoning: reactive approaches

- Dominating paradigm: Markov Decision Processes (MDPs)
- Essentially stochastic finite-state machines with utility-based definition of design objective and preferences over states (and actions)
- Simple, domain-independent model, applied to many domains across AI
- Can be seen as a continuation of work on reactive architectures
  - subsumption architecture, situated automata, etc

Single-agent reasoning: reactive approaches

- Strengths:
  - Strong connections to control engineering, stochastic optimisation, machine learning
  - Well-defined optimality criteria (expected utility maximisation) with strong decision- and probability-theoretic grounding
  - Well-understood solution mechanisms (at least for single-agent/cooperative setting) with formal convergence guarantees
  - Works under uncertainty, partial observability (POMDPs), noise and failure
  - Easily combined with game-theoretic concepts (=stochastic/repeated games), and learning (reinforcement learning machinery)

Single-agent reasoning: reactive approaches

- Weaknesses:
  - Tractability problems in realistic state spaces, hierarchical approaches exist but are not straightforward
  - Poor in terms of transparency, human readability, modularity, extensibility
  - Not easy to integrate seamlessly with other components in principled way
  - Convergence “in the limit” often not very useful in real-world problems
  - Performance sensitive to state-space definitions and reward function design
  - Difficult to formulate performance criteria for strategic multiagent systems settings
  - Hard to debug in case of failure

Single-agent reasoning: deliberative approaches

- Dominant paradigm: Belief-Desire-Intention model of practical reasoning
- Based on Bratman's model of human practical reason:
  - Beliefs determine how desires (long-term preferences of agent) are transformed into intentions (concrete adoption of plans in continuous loop
- Distinction between deliberation (commitment to ends) and means-ends reasoning (commitment to means)
- Intention theory enforces rationality constraints: achieved/impossible goals are dropped, intentions must not contradict each other, are believed to be possible etc

Single-agent reasoning: deliberative approaches

- Planning and execution interleaved with deliberation and meta-level reasoning (intention reconsideration)
- Essentially a method for reactive (continuous) planning, where existing intentions and plans together with external events trigger new intentions
- Planning often based on pre-computed plan libraries not run-time automated plan synthesis
- Paradigm emerged from knowledge-based AI and planning
- Predecessors: production systems, SOAR, early agent programming languages (AGENT0, MetateM etc)

Single-agent reasoning: deliberative approaches

- Strengths
  - Knowledge-level/mentalistic concepts appeal to human intuition
  - Powerful means of balancing reactive and proactive behaviour
  - Result in modular, transparent, extensible architectures
  - Enable use of domain-specific background knowledge
  - Exploit the power of deductive reasoning and knowledge-based planning (to varying extents)
  - Relational representations are much more succinct than enumerated ones (and come with associated complexity and tractability problems)
  - Gave rise to a plethora of implemented reasoning engines and agent-based programming languages (AgentSpeak, 3APL, Jason etc)

Single-agent reasoning: deliberative approaches

- Weaknesses
  - No domain-independent performance metrics or guarantees
  - No notion of optimality or clear problem formulation, no “solution” algorithm
  - Fairly coarse-grained notions of correctness (e.g. formal properties of intention logics)
  - Meta-reasoning hard to relate to decision-theoretic notions (goals are less expressive than utilities)
  - No explicit (quantitative) account of uncertainty and partial observability
  - Extension to multiagent architectures or social reasoning not obvious (though many suggested models)

Methodology intermission

- First example of a deep methodological issue: **"Formal" vs. "informal" approaches**
- "Formal" approaches:
  - Rigorous formulation of problem and solution
  - Formal success criteria and performance metrics
  - Domain-independent, broadly applicable frameworks
  - Solution methods with performance guarantees
  - Limited expressiveness and real-world adequacy
- "Informal" approaches:
  - Intuitively appealing and ontologically richer
  - Give more guidance in terms of design and practical use
  - Leave smaller gap between model and real-world problem
  - Resulting systems easier to understand and analyse
  - Leave more space for identifying interesting sub-problems

Methodology intermission

- Classification has to be taken with pinch of salt
  - Lots of formal work on BDI (but not so important for everyday practice of BDI)
  - Lots of extensions to basic MDP model (but hard to analyse)
  - Hybrid approaches exist (but are poorly understood)
  - Most researchers agree that both sides are needed
- Major questions to decide on appropriate method
  - How well-understood is the problem we want to tackle?
  - How domain-independent do we need the solution to be?
  - What “dirty” real-world issues do we have to deal with?
- We will revisit to these issues in part III

Single-agent reasoning: the state of the art

- Increasing interest in "sub-symbolic"/numerical methods mirrors trend in other areas
  - reinforcement learning, robotics, uncertainty communities
- A lot of work on this side currently focuses on multiagent versions of the problem
  - decentralised (PO)MDPs, multiagent reinforcement learning, abstractions, mechanism design
- Most work on deliberative architectures these days concerned with extensions/variations
  - social reasoning, communication, programming languages, applications
- Long-term vision to combine the two (fundamental question in AI)
  - Many believe "the time has come" (limits reached, computational power now available, etc)

Multiagent reasoning: cooperative systems

- Setting: decentralisation of information and control when computational components are agents
- Cooperative and benevolent agents
  - goal is ultimately shared, no danger of malicious behaviour, no conflict of interest
- Dominating paradigm: "teamwork" among collaborating teammates
- Concerned with all stages of distributed problem solving
  - team formation, agreement on goals, agreement on means, joint execution and monitoring

Multiagent reasoning: cooperative systems

- Research issues:
  - How to ensure consistency of global state (joint intentions framework, partial global planning, blackboards)
  - How to allocate tasks and resources (contract nets, coalition structure generation)
  - How to deal with task execution and failure (communication mechanisms, joint plan repair and replanning etc)
  - How to combine partial results

Multiagent reasoning: cooperative systems

- Strengths
  - Frameworks address all aspects of interaction among intelligent agents in real-world systems
  - Thinking of real-world domains, cooperative modelling is often appropriate (organisations, robot teams etc)
  - Many very impressive applications of multiagent systems fall into this category! (And have done so for many years!)
- Weaknesses
  - Often "big" models, hard to reproduce work described
  - Work in this area very heterogeneous (except for MDP-like settings)
  - Multiagent planning, in particular, is extremely fragmented and lacks small set of core problems
  - Seen by many as a limited sub-category of systems that allow self-interested agents

Multiagent reasoning: non-cooperative systems

- Generalisation of previous scenario allowing self-interest
  - This small change makes a world of difference!
- Main research themes:
  - How to behave in the presence of other self-interested agents to further one's goals
  - How to reach agreement in the presence of conflict of interest
- Area stands in tradition of game theory
  - The two areas two roughly correspond to non-cooperative and cooperative game theory
- Usually relies on "small" mathematical models of agents, actions, outcomes and preferences (often numerical utilities)

Multiagent reasoning: non-cooperative systems

- Can be broken down into more concrete sub-topics:
  - Strategic behaviour: how to act given different preferences of others
  - Resource/task allocation: how to distribute goods given agents' preferences
  - Coalition formation: how to form groups and split the gain obtained from collaboration
  - Social choice: how to make group decisions given everyone's preferences
  - Bargaining: how to negotiate effectively in the presence of conflict of interest
- Computational focus (unlike standard game theory)
  - algorithms for mechanism design, equilibrium computation, coalition structure formation, winner determination in auctions, representation languages

Multiagent reasoning: non-cooperative systems

- Strengths
  - Mathematically rigorous, well-defined problems
  - Clearly distinct research focus from more “fuzzy” areas
  - Introduced radically new model of non-cooperative computation (a genuine agents research contribution!)
  - Cross-fertilisation with other domains (e.g. theoretical computer science, combinatorial optimisation)
- Weaknesses
  - Models often limited and based on heavy assumptions
  - Many negative results or only theoretical analysis (e.g. worst-case complexity)
  - Often actually about “avoiding” agent reasoning
  - This often implies lack of guidance for agent design in real-world scenarios
  - Results say little about fully-fledged agent applications

Multiagent reasoning: the state of the art

- Non-cooperative approach highly popular (beyond agents as well)
  - Areas like auctions commercially highly significant
- Dichotomy may be less clearcut that assumed
  - Many real systems “mostly” cooperative with some element of self-interest
- Combination of both approaches would be useful
  - Even in cooperative settings self-interest could be used to improve efficiency
  - Programs with preferences usually imply “explosive” complexity
- Automated reasoning about games underdeveloped
  - Virtually all analysis still done a priori by human expert

Multiagent approaches to standard AI problems

There is also much work on applying an “agents” approach to more traditional AI problems:

- Distributed search
- Distributed constraint satisfaction
- Distributed knowledge representation
- Distributed theorem proving
- Distributed planning
- Distributed game-playing
- Distributed machine learning
- Distributed sensing and acting

Multiagent approaches to standard AI problems

Criticisms of these approaches

- Often accused of mere computational distribution, “agency” doesn’t add any value
  - danger of reinventing the wheel
- Methodological pitfall: potential lack of novelty
  - when problem becomes narrow enough and well-defined, it might look like a standard AI problem
- But there are also genuinely novel problems here!
  - multiagent reinforcement learning, planning games, argumentation and knowledge integration
- Safe way to extend an existing problem is to go non-cooperative (decentralisation helps less)
  - but need motivating scenarios, and generally problems become very hard then

Multiagent interaction and coordination methods

- Agent communication languages, protocols, semantics, ontologies
  - A lot of standardisation work in 90s and agent plaftorms
  - Inherent problem: addresses only part of a whole MAS
  - Considered “done” by many researchers, a standard part of applied MAS design
- Norms, institutions, and organisations
  - System-level models of behavioural expectations as a control mechanism
  - Mostly modelling, logical specification, few algorithms
  - Also “small” social laws, including logic-based variant
  - Biased toward “regimentation” of open systems (tension!)

Multiagent interaction and coordination methods

- Trust and reputation
  - Dominant paradigm for mutual modelling in open MAS
  - For some just a multiagent learning problem
  - For others lots of "human" aspects
- Argumentation
  - Integration of conflicting knowledge, non-monotonic reasoning tradition
  - Dominant approach for logic-based dealing with uncertainty
  - Little account of strategic behaviour and practical algorithms

Agent systems engineering

- Agent-oriented software engineering (beyond AI agenda)
  - Methodologies, programming languages, tools, verification
- Mostly based on deliberative models and declarative programming
  - Much current work part of research on BDI
- Main contribution: agent-style programming paradigm
  - Not established as mainstream programming paradigm
- Usual problems of evaluation
  - Needs large community to assess practical value
  - Multiagent programming contest is useful

Human-oriented agent systems

- Also outside standard AI agenda
- Virtual agents
  - User assistants & tutors, storytelling, believable behaviour
  - Related to animation, user modelling, intelligent tutoring
  - Usually rather applied and interdisciplinary
- Social simulation
  - Use agent-based methods for social science
  - Usually large-scale systems, simple agents
  - Problem of validity of artificially generated data
- Hard to identify domain-independent performance measures, require human-based validation (often qualitative)

Summary: The State of the Union

- Very high diversity within the area, many sub-areas lack clear problem and solution formulations
  - A lot of work overlaps with other fields
- Fundamental barriers to cross-fertilisation
- Few “reference systems” within which to test new components
  - Multiagent mechanisms assume “given” agents
- Several attempts to introduce benchmarks through competitions
  - RoboCup (+Rescue), ART, TAC/CAT, progr. contest
- Novel application areas raise new challenges
  - electronic markets, social networks, human-based computation, mobile computing

How do we evaluate agents research?

Part III: How do we evaluate agents research?

Introduction

- Before we discuss research design, we need to clarify objectives
- Achievement of objectives requires judgement of outcomes, should determine research approach
- Start by clarifying how we can evaluate agents research
    - Through formal modelling and analytical proof
    - Through experiments with prototypical systems
    - Through experiments with people
    - Through description of concrete applications
    - Through interpretation of observations
- Will not say much about mix of these, but often pursued

Theorems

- Show that a formal system has some property by way of proof
- Produces new knowledge about some mathematical truth
- Strictly speaking not concerned with the so-called “real world”
- In agents research should at least have some computational consequences
- Requires models to be fully formalised and sufficiently simple

Theorems

- Common types of proof encountered in agents field:
  - Soundness and completeness (logic)
  - Complexity (mechanism design, logic)
  - Convergence (optimisation, learning)
  - Correctness (algorithms)
- Many based on worst-case or “in the limit” arguments
- Idea of systematic progress of understanding of formal structures

Theorems

- Strengths
  - Clarity, rigour, elegance, indisputability
  - Domain-independence (within limits of formal structure)
  - Incrementality/non-redundance: theorems have to be new
  - Esteem: many famous researchers had big theorems ("theory envy")
- Weaknesses
  - Difficulty not necessarily indicative of significance
  - Operate within limited, assumption-laden models
  - Hard to incorporate real-world aspects, esp. humans
  - Temptation to underestimate "social process" of science

Simulation

- Implement a system that simulates the real phenomenon
- Use quantitative measurements on this system to verify some property
- Important to distinguish between real world and simulation system
    - Implicit assumption that simulation is similar to reality of a more general class of systems
- Based on empiricism: objective observation of events as a basis to verify truth
- Strong relation to probability theory and statistics

Simulation

- Common simulation-based approaches in agents:
  - "Toy domains" (minimal problems that have required properties, often reused within a community)
  - Simplified simulation of one or more real-world application domains (usually leaving out details)
  - Benchmarks (allow comparison to other approaches)
  - Reference implementations (allow fast integration)
  - Social simulation (agents assumed to behave like people)
- Main methodological issue: coverage
  - Regarding problem/solution parameters, realism of example domains, impossibility of exhaustive coverage

Simulation

- Strengths
  - Independent of level of formalisation
  - (Potential for) Realism
  - (At least prototypical) implementations of methods
  - Emphasis on computation
- Weaknesses
  - Many decisions regarding specific evaluation strategy
  - Often not directly verifiable for other researchers
  - Rarely rigorous statistical significance tests
  - Aftertaste of ambiguity, no “hard” results

Human experimentation

- Systems that interact with or supposed to behave like humans
  - One should expect that this is validated with humans
- Common approaches:
  - Using human-collected datasets as system input
  - Small-scale user trials (observational techniques)
  - Large-scale remote user trials (e.g. AMT)
  - Qualitative empirical research
  - Participatory design (e.g. early adopter groups)
- Can be quantitative (based on statistical analysis) or not

Human experimentation

- Strengths
  - Human-in-the-loop, expose systems to full real-world complexities
  - Indispensable for validating certain properties that cannot be simulated
- Weaknesses
  - Results highly sensitive to experiment design
  - Qualitative research hard to objectify
  - Even greater aftertaste of ambiguity
  - Costly, time-consuming (and dull for many?)

Case studies

- A variation of experimental methodologies (human or not)
- But no claim to statistical significance or coverage beyond scenario analysed
- Often used when proposing complex artefacts and methodologies
- Common in programming languages, software engineering, applications
- Strength: going “all the way”
- Weakness: plagued by the “instance trap”
    - What claims can be made beyond a specific example?

Interpretative approaches

- No formal/experimental/system-building evaluation is performed without human interpretation
- Hermeneutic view: truth unfolds within context of experience of those who are reasoning about it
    - Heidegger: “Physics tells us how electricity *behaves*, not what it *is*.”
- Unusual (undesirable?) outlook for a “positivist” science like computing/AI
    - But deeply embedded in the area, maybe more so than in many other others (social science influence etc)
- Commonly observed pitfall: model concepts (or claim to observe them – emergence) borrowed from other discipline, not making any link to other computational methods

Presentation

You also have to present your results well . . .

- Important to consider social process of science
  - Often what others will find acceptable guides evaluation strategy
- The reviewer is always right!
  - Underestimating problems that can arise from miscommunication
- Hard to find a "seminal" paper that is not extremely well-written
  - Technical writing is an art in its own right
- Very common problem in agents: evaluation methodology not actually discussed or justified (most people think result itself is main thing)

Summary

- Theorems vs. experiments mirrors “small” vs. “big” dichotomy (some would say “clean” vs. “dirty”)
- There are no simple recipes, element of risk
- Any evaluation involves interpretation by humans
- Commonly observed problems with evaluation in agents research:
    - “Human”-like properties not verified against humans
    - “Solipstistic” evaluation of an approach against itself
    - Lack of critical assessment of contribution in theory papers

Part IV: How should we do agents research?

Introduction

- Obviously there isn't one single answer
  - Science operates on frontier of knowledge, how to do it will always be subject to debate
- Previously we saw what main directions and methods can be followed
- Now I will highlight a series of methodological issues that are worth considering
- A good answer to these doesn't guarantee success, but it is useful to avoid failure

A reason for research

- There should be a solid motivation behind any research
- Computing research produces
  - Knowledge (theorems, analyses, method(ologie)s)
  - Artefacts (architectures, implementations, tools, data)
- So the first question is? Which of these will be produced, and why?
- A good indication is activity in an area, shows there is interest in a problem
  - Taking an existing, well-understood problem makes things even easier
- Demand regarding a real-world application gives more applied outlook
  - But much theoretical work abuses this, is never intended to really contribute

Back and forth

- Tension between expertise-driven and goal-driven approach
  - We are asked to motivate research its objective
  - But we are actually interested in moving “forward” from existing knowledge
- Practical reasons for “forward” view:
  - Novel, good research questions appear out of the context of existing work
  - Socially more effective if experts keep working on the things they know best
- Pragmatic solution: Work incrementally from state of the art, but present research as if motivated by research goal
- However it is important to stop and think “out of the box” from time to time

Good and bad hypotheses

- In an ideal world, every research hypothesis would be verifiable and falsifiable
  - This means the hypothesis must specify the method of verification!
- In the real world, often the hypothesis is not even explicitly stated
  - Test: take a random paper and try to find clear statement
- Problematic: “new” approaches that do something that hasn’t been done before
  - When would this be a success? And what would it prove?

Good and bad hypotheses

- In reality often results not conclusive, work is preliminary, objective is vague
  - Understandable, but strong contributions have less of that
- Vanilla-flavour hypothesis: The method presented performs better than the best previous method(s) along some dimension
- Performance measure can be
  - quantitative, e.g. speed, convergence, efficiency, predictability, robustness, stability
  - non-quantitative, e.g. expressiveness, simplicity, elegance, generality, usability

Good and bad hypotheses

- Many non-quantitative ones are not objectively verifiable
  - At least not unequivocally, presentation requires discursive argumentation
- Tricky: how good can we reasonably expect the next improvement to be?
  - From ε-improvement via “normal science” to seminal contributions and “paradigm shifts”
- For exploratory work, good to present research as step *toward* real hypothesis
  - “Pilot study” idea, small-scale study preceeds full-scaled experiment

In search of rigour?

- Discussed “theory envy” before, mathematical truth is considered “king”
  - Traditionally considered intellectually challenging, elegant, general
- But more realistic models and systems usually lead to fewer hard results
  - Complex models harder to analyse formally
- Conversely, more theoretical work is often less significant in practical applications
  - The exceptions to this rule are the real gems in computer science! We have a couple of those in agents too!

In search of rigour?

- The question is: how “dirty” do we want to/have to go?
- Tension between laziness/complacency and avoiding theory envy
  - Search for closest possible formal model(s)
  - If it cannot be extended to accommodate needs, justify specifically
  - Attempt to formulate distinct novel “well-defined” problems
- Pitfall: Dismiss existing formalisms, build something more complex, be forced to apply “ad hoc” solution
- Contributions without formal foundation can also be very influential (but harder to ensure generality)

Reinventing the wheel?

- A clearly visible problem in agents: developing a novel approach
  - In particular, not thinking hard enough about similar ideas in other fields
- Incremental research is more tedious, but more rewarding
  - Provides clear path to evaluation
  - Ensures there is interest in the area
  - Methodological problems are already resolved
- A practical idea: pick highly-cited technical papers, and continue their line of research
  - Also seems more beneficial for progress of science at macro-level

The instance trap?

- Applied research is highly valuable but requires more commitment to a particular scenario
- This is fine, as long as specific artefact is objective of the research
- Dangerous however, when a claim is made to generality of result
  - One domain can never be enough
  - Lots of specific (necessary) choices will “obscure” principles
  - Results cannot be reproduced without big effort
- In agents, applied research sometimes comes in the guise of fundamental research (including my own work)
- . . . worse still, without producing a real fully-fledged application!

The enemy never sleeps

- Many research problems addressed in agents research are also of interest to others!
  - Currently especially anything related to game theory
- Important to be aware of other areas, especially more established ones
  - Not only competition but cross-fertilisation possible
- "Entrenchment" in own views is a common problem of communities to be wary of!
- Is is not enough to say "we do things differently"
  - One has to prove that, and show that it has some advantage
- Ultimately, any agents research contribution has to be defendable toward a general (at least) computer science audience!

In the interest of humanity

- How can one achieve maximal impact of own work?
- Pick popular topics, but ensure long-term vision
- Make sure you know your audience
- Publish in the best outlets that will accept your work
- Talk to people who do similar work
- Get involved with the community

Summary

- A tour of various methodological issues in agents research
- Addressed choice of objectives, methods, evaluation
- How to avoid common pitfalls
- How to maximise research impact
- Attempted to contextualise agents research withing “big” picture of computing and science more generally

Conclusions

Conclusions

Conclusions

- Agents and multiagent systems: a very diverse, inclusive, dynamically changing field
- Characterised by mix of formal and informal, theoretical and applied
- A variety of diverse approaches, “big achievements” of area not always clear
- Hard to get a big picture of the whole field, increasing fragmentation
- Methodological problems that arise from these issues

Conclusions

Some personal thoughts

- An exciting field to be in, with very open-minded people
- Sometimes dangerous temptation to cover everything
- Tension between “small” and “big” approaches here to stay, more work should be done on combination
- We have the right tools for future computer applications
- It is time we deliver the technology that shows that!

The End

Thank you for your attention and participation!