Introduction

COMP90049 and COMP30018 Knowledge Technologies

Justin Zobel and modified by Rao Kotagiri

Semester 1

THE UNIVERSITY OF MELBOURNE

Subject overview from handbook

“Much of the world’s knowledge is stored in the form of unstructured data (e.g., text) or implicitly in structured data (e.g., databases).

“In this subject students will learn algorithms and data structures for extracting, retrieving and storing explicit knowledge from various data sources, with a focus on the web.

“Topics include: data encoding, web crawling, clustering, regular expressions, pattern mining, Bayesian learning, instance-based learning, document indexing, database storage and indexing, and text retrieval.”

On successful completion of the subject, students should be able to:

- “Describe and apply fundamentals of knowledge systems, including data acquisition and aggregation, knowledge extraction, text retrieval, machine learning and data mining.”

What you should gain from this subject

- Exposure to a range of computing technologies for:
  - Making use of uncertain, irregular, or complex data.
  - Accomplishing tasks that may not be well-specified or well-understood.
  - Supporting humans who are engaged in discovery or decision-making.
- A broader understanding of the kinds of things that can – and can't – be accomplished computationally.
- Insight into some research activities in computing, why they are undertaken, and how.

Uses of computation

What is computation for?

Much of computer science concerns our attempts to coerce (instruct) computers into accomplishing *tasks*, loosely definable as:

- An identified source of data.
- An identified context or situation.
- A desired outcome.

The data may be created for the task, or might be derived from the physical world – transformed, by a device, into bits from entities or events in our universe.

A context might be a specific piece of hardware or operating system, or might be assumptions such as “the numbers represent prices” or “the text is in ASCII”.

An outcome might be a number, an action, a list of results, . . .

Uses of computation

Computers and algorithms were originally developed to solve what might be called *concrete* tasks. For example (tiny selection):

- Compute a missile trajectory.
- Crack a code (decryption without the decryption key).
- Do accountancy over financial data.
- Operate a camera (focus, exposure), store the image.
- Guide a cutting tool, operate an assembly line.
- Map mouse movements to cursor movements.

In common: the task is well-defined, we can assess whether the solution is correct.

In these tasks, the data is transformed in a mechanical way or leads to a mechanical action, but only in a very limited way do they enhance our (that is, human) knowledge.

Hence – not generally considered “knowledge technologies”. In Knowledge Science such problems are called deterministic problems!

Knowledge Management and Science View

All problems can be classified into 5 categories:

- Simple, in which the relationship between cause and effect is clear, the approach is to Sense - Categorise - Respond and we can apply best practice. These problems are deterministic in the sense of solving.
- Complicated, in which the relationship between cause and effect requires analysis or some other form of investigation and/or the application of expert knowledge, the approach is to Sense - Analyze - Respond and we can apply good practice. These problems are solved using Knowledge Technologies.
- Complex, in which the relationship between cause and effect can only be perceived in retrospect, but not in advance, the approach is to Probe - Sense - Respond and we can sense emergent practice. Learn from these experiences.
- Chaotic, in which there is no obvious relationship between cause and effect at systems level, the approach is to Act - Sense - Respond and we can discover novel practice.
- Disorder, in which is the state of not knowing what type of causality exists, in which state people will revert to their own comfort zone in making a decision. We may not learn much from this experience.

Knowledge Management and Science View

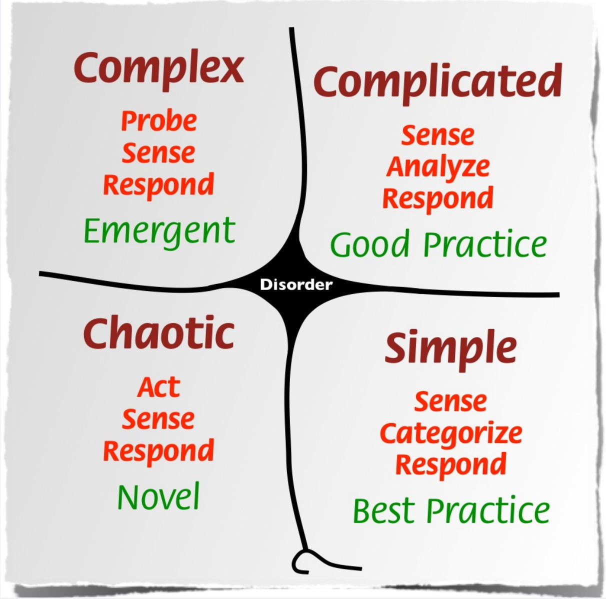

Figure: Problem Space

Knowledge tasks

Consider tasks where the data is irregular or unreliable, or the outcome is not well-defined:

- Compression of an image.
- List of answers to a typical web query.
- Translation between languages.
- Identification of what a health condition might be caused by; identification of a treatment.
- Finding an "optimal" route between two locations. (Optimal? Distance, time, stress, fuel?)
- Deciding what movie to watch.

"What movie to watch?" (Or music to buy, or place to visit, or . . .) This is not a computational task – but in such tasks we do use computers to *mediate* between us and data, in helping to reach a decision.

Context is critical: the origin of the data, the consumer of the output.

These use, produce, or enhance human knowledge.

Knowledge task example

Why is translation between languages not well-defined?

Because translation leads to loss of meaning, and loss of nuance (subtle difference) and "feel". Test: ask a human to translate from English to some other language; ask another human to translate back. Justin tried via Chinese: (English $- >$ Chinese $- >$ English)

In: *Consider tasks where the outcome is not well-defined*

Out: *Consider work where the result is unclear*

Another example:

In: *The spirit is willing but the flesh is weak*

Out: *The vodka is good, but the meat is rotten*

Why would we expect a machine to do better? What *is* a correct translation?

Paraphrasing Julian Barnes (London Review of Books, 18 Nov 2010), imagine reading a famous 1850s French novel in English. What do you want? Probably, that it provoke the same reactions in you as in a French reader; but what about the topical references that only a French speaker would know?; or the glaring errors concerning English culture that a French speaker wouldn't notice; and what English? An attempt at 1850s English, with disused forms of expression, or modernized? And what judgements about class and education? (The two societies were not equivalent.) Are trousers held up by braces, or pants held up

Characteristics of knowledge technologies

These technologies tend to be either fairly general (e.g., machine learning) or fairly specific (e.g., machine translation).

General: the data must be transformed to suit the axioms or assumptions of the method, in a rigorous way.

Specific: detailed understanding of the task is used to drive development of the method, perhaps by drawing on a toolkit of components and of solutions to similar problems.

- A specialized problem: parse a particular language.
- An approximate problem: assign a document to a category.
- A general problem: find features of the data items that discriminate between categories.

A broader perspective

Knowledge technologies are occasionally transformational (disruptive technologies)– consider social networking, web search.

Innovations of this kind are often designed to assist or augment an existing activity, but the consequence is that they displace it entirely. Think of:

- The impact of social networking on email.
- The impact of search engines on libraries and encyclopaedias.
- The impact of blogging (and tweeting) on reportage and newspapers.

Computational thinking

Finding solutions to tasks requires application of computational thinking:

- How should the data be represented?
- How should it be manipulated?
- What heuristics or simplifications can be safely applied?
- Can the problem be transformed or rearranged in a way that usefully changes the costs, e.g., space and time complexity?
- Does it have properties that let it be addressed by sorting?
- Does it have properties that let it be addressed by searching?
- Is it possible to eliminate the need to consider global properties, allowing a focus on local properties? That is, does all of the data have to be considered holistically, or can it be divided in some way?
- How will a solution behave as the data approaches boundary conditions? (Increase or decrease in number of errors; data items unique or frequently repeated; as item size or item number grows, . . .)

Thinking tools for knowledge technologies

Consider *effectiveness* rather than *correctness*. (Can a document ranking possibly be “correct”?). Sometimes even correctness cannot be defined and/or asserted!

Identify features and characteristics that can be quantified.

Identify approximations to the task.

Consider whether the outcome is likely to seem plausible or appealing.

Whether it makes sense to consider training data from which tailored solutions can be automatically learnt. (Which may make a solution easy, but may make it difficult to gain insight into the problem.)

Ask: What does signal look like? What does noise look like? What would a human do, given sufficient stamina and memory? What output would a human produce?

Is a human part of the loop in some way? How is the output to be consumed?

Example: All of these questions apply to aspects of web search.

Content

Main topics:

- Pattern and string matching, spelling correction
- Basic text processing, web and text search
- Machine learning
- Clustering, classification
- Data mining

Along the way:

- Measurement of effectiveness
- Insights into current research
- Bayesian reasoning
- Some interesting algorithms, a little theory

Beyond the scope of this subject

Far more knowledge technology topics are out than are in!

- Computational modelling: traffic, medical, climate, . . .
- General approximation and reasoning techniques for computing solutions in the presence of formal intractability. (But we do look at a couple of specific examples.)
- Natural language processing, machine translation.
- Image analysis, image matching.
- Theorem proving.

. . . and many others.

Who and where

Lecturer:

Prof Rao Kotagiri, Doug McDonelll Buiding Room 7.10
kotagiri@unimelb.edu.au

Head Tutor:
Mr Jeremy Nicholson nj@unimelb.edu.au

Lectures:

COMP90049 L01/01 Monday 10:00am- 11:00am 1 hour ERC-132 (Charles Pearson Theatre)
COMP90049 L02/01 Tuesday 4:15pm - 5:15pm 1 hour Doug McDonell 103(Herbert Wilson Theatre)

Who and where

Workshops commence from 2nd week:

There are 10 workshops and you have to choose one of them.

Website: `www.lms.unimelb.edu.au`; use `unimelb` .

Prerequisites

Subjects:

- COMP20003 (433-253/298) Algorithms and Data Structures
- COMP90038 (433-521) Algorithms and Complexity

Skills:

- Data structures & algorithms coding in C, C++, Python, Java or similar.
- Assignments to be completed in C, C++, Python, or Java. (Elementary C and scripts to be used in lectures.)
- Familiarity with formal mathematical notations.
- Basic understanding of statistics and information theory helpful but not essential.

Intended schedule

Week	Day	Content
1	Monday (27 Feb) Lecture 1	Introduction
	Tuesday (28Feb) Lecture 2	Approx. matching
2	Monday (6 March) Lecture 3	Approx. matching
	Tuesday (7 March) Lecture 4	Approx. matching
3	Monday (13 March) Lecture 5	Approx. matching
	uesday (14 March) Lecture 6	Text processing
4	Monday (20 March) Lecture 7	Information Retrieval
	Tuesday (21 March) Lecture 8	Information Retrieval
5	Monday (27 March) Lecture 9	Web search
	Tuesday (28 March) Lecture 10	Introduction to Data Mini
6	Monday (3 April) Lecture 5	Web search
	Tuesday (4 April)) Exam	*Mid-semester te*
7	Monday (10 April) Lecture 5	Introduction to Data Mini
	Tuesday (11 April) Lecture 6	Introduction to basic Pro
Easter	*break – No lectures 14April*	*to 23 April*
8	Monday (24 April)) Lecture 11	Introduction Machine Lea
	Tuesday (25 April) Public Holiday	

Intended schedule

Week	Day	Content
9	Monday (1 May))Lecture 12	Classification-1
	Tuesday (2 May) Lecture 13	Classification-2
10	Monday (8 May))Lecture 12	Classification-3
	Tuesday (9 May)) Lecture 14	Classification-3
11	Monday (15May) Lecture 15	Clustering
	Tuesday (16May)) Lecture 16	Data Mining
12	Monday (22 May)Lecture 20	Data Mining
	Tuesday (23May) Lecture 19	Data Mining

Texts and references

There is no prescribed text.

- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008), *Information Retrieval*, Cambridge University Press.
  Freely available at `informationretrieval.org`
- Pang-Ning Tan, Michael Steinbach and Vipin Kumar (2005) *Introduction to Data Mining*, Addison-Wesley.

Other readings: Provided as links to conference and journal papers. You are not expected to master every element of every paper, but you are expected to broadly understand the subject material.

Assessment

Comp90049

40%	2 × assignments Expected to require around 36 hours Accuracy, novelty, written presentation
10%	Mid-semester test
50%	Final exam

Assessment

Projects are undertaken individually. No group work.

Hurdle requirements: to pass the subject you must achieve at least a pass in the assignment component (i.e., $\geq 20/40$) and the test+exam component (i.e., $\geq 30/60$). Failing either component means a fail in the subject overall.

Originality: All submitted projects must be your own work. Effort will be made to verify that your work is original. Submission of copied materials can have serious consequences.
See `academichonesty.unimelb.edu.au` and `academichonesty.unimelb.edu.au/plagiarism.html`.

Help!

I'm available for consultation in my CIS ((Doug McDonell Building 7.10) office 11:00am–12:00am Monday , 9:00-10:00 Tuesday each week (except for a couple of absences) and before and after lectures.

Otherwise, make an appointment – use email.

Questions by email are welcome . . . but I don't promise to answer in less than 24 hours. I may post answers on the LMS if I think they are of general interest.

Feel free to use discussion forums, etc., on the LMS.