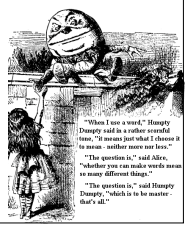
'intelligent' interaction and exploration

Alan Dix
Lancaster University, UK
www.hcibook.com/alan/teaching/delos

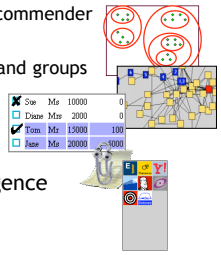
note ...

- using 'intelligent' to include
  - traditional artificial intelligence
  - statistical methods
  - information retrieval techniques
  - neural nets
  - genetic algorithms
  - simple heuristics



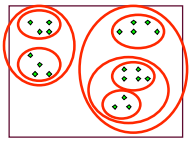
overview

- clustering and similarity-based techniques
  - using attributes, text, recommender
  - types of algorithm
  - interacting with clusters and groups
- precise intelligence!
  - Query-by-Browsing
- IUI and appropriate intelligence



clustering

- hierarchical visualisation
  - uses human allocated groups: taxonomies, keywords etc.
- can also automatically group
  - based on some measure of similarity
  - called 'clustering'
  - can be single level
  - or hierarchical



similarity measures

- based an attributes:
  - author, size, etc.
- based on content
  - words, images, etc.
- based on usage
  - who has seen it
  - when it was seen
- based on external structures
  - web links (google), RDF

attribute-based distance

- categorical attributes (e.g. colour)
  - similarity(d1,d2) = number of attributes the same
- continuous attributes
  - use (weighted) Euclidean distance
  - distance(d1,d2) = $\sum_{\text{attributes } a}$ (d1.a - d2.a)^2
  - similarity is 1/distance or maxdist-distance
- combine for mixed attributes
  - weighted sum of distance/similarity, max distance

content-based similarity - text

- see 'information retrieval' literature
- use measures based on co-occurrence of words

simple measure:
given two documents d1, d2
let C = number of words in both d1 and d2
D = number of words in just one document

similarity(d1,d2) = 2C / (D+2C)

content based similarity example

d1 = "digital libraries clustering using IR"
d2 = "a clustering visualisation of a digital library"

C = words in common = 2
D = different words = 8

similarity(d1,d2) = 2C / (D+2C) = 4/12 = 33%

content based similarity stop words and stemming

- stop words
  - usually ignore common words
    e.g. a, of, and, ...
- stemming
  - treat words with a similar 'stem' as the same:
    e.g. library == libraries
  - do this by chopping words to their 'stem':
    library => 'librar', clustering => 'cluster'

content based similarity example revistited

d1 = "digital libraries clustering using IR"
d2 = "a clustering visualisation of a digital library"

remove stopwords:
d1 = "digital libraries clustering using IR"
d2 = "clustering visualisation digital library"

stemming:
d1 = "digit librar cluster using IR"
d2 = "cluster visual digit librar"

C = 3, D = 3
similarity(d1,d2) = 2C / (D+2C) = 6/9 = 66%

content based similarity further refinements

- may weight some words more:
  - words in titles and headings
  - words near beginning of document
  - less common words (e.g. 'entropy' more than 'library')
    (need some large lexicon for this)

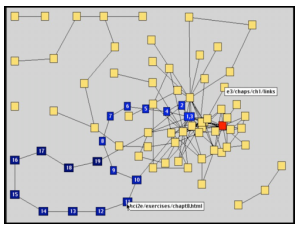
  C = sum of weights of words in common
  D = sum of weights of all words in documents
- may use more complex formulae:
  - χ squared, entropy, etc. ... a 'black art'!!

recommender systems

- use other people's past behaviour
- simple popularity of access
  - how often they view a document, follow a link
- user ratings (star rankings etc.)
- similar users (based on patterns of use)
  - Amazon "other people who bought this book ..."

recommender example: web path-based visualisation

- take web logs and extract sessions
- pages similar if they are often next to each other in session logs
- draw pages close if they are 'similar'
- frequent paths come out close



clustering and layout techniques

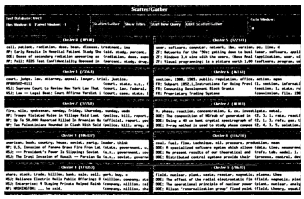
- traditional statistical
  - k-means - hierarchical clusters
  - factor analysis, multidimensional scaling
    - creates dimensions for 2D scatter layout
- neural nets
  - ART (adaptive resonance theory)
    - single level clusters
  - Kohonen nets (self-organising maps)
    - clusters and creates 2D organisation on grid

using clusters
the scatter/gather browser

take a collection of documents
scatter:
- group into fixed number of clusters
- displays clusters to user

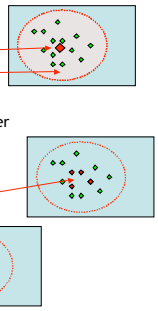
gather:
- user selects one or more clusters
- system collects these together

scatter:
- system clusters this new collection
- ...



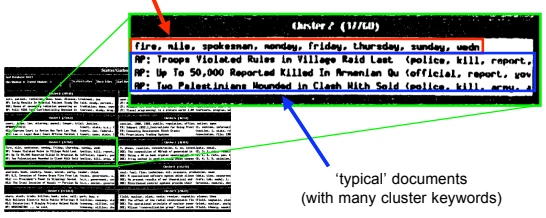
displaying groups/clusters

- numeric attributes
  - use average or region
- categorical attributes
  - show values of attributes common to cluster
- text and images
  - no sensible 'average' to display
  - use typical documents/images
  - central to cluster ... or spread within cluster



displaying clusters
scatter-gather browser

keywords (created by clustering algorithm)



'typical' documents
(with many cluster keywords)

precision

- similarity clustering or 'others like this'
  - usually approximate algorithms
  - usually no explanation
- good if there are lots of 'good enough' matches
- less good if you need to be sure of what you have

Query by Browsing

- user chooses records of interest
  - ✓ tick for those wanted
  - ✗ cross for those not wanted
- system infers query
  - web version uses rule induction
  - variant of Quinlan's ID3

Data

	Name	Title	Wage	Overdraft
✗	Fred	Mr	12000	500
✓	John	Dr	20000	10000
✗	Sue	Ms	10000	0
☐	Diane	Mrs	2000	0
☐	Tom	Mr	15000	100
☐	Jane	Ms	20000	-5000
☐	Dick	Mr	10000	50

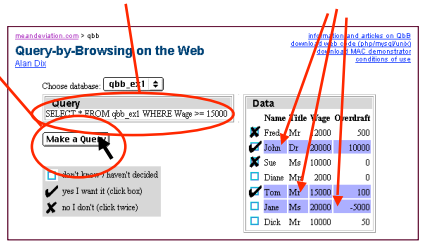
www.meandeviation.com/qbb

Query by Browsing what it looks like

user asks system to make a query

system infers SQL query

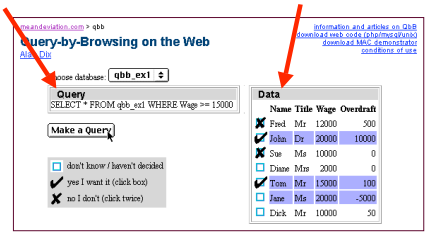
query results highlighted



Query by Browsing dual representation

query (intensional) for precision

listing (extensional) for understanding



intelligent user interfaces

- have had a bad history!
- why?
  - often focused on clever techniques
  - forget wider interaction

e.g. intelligent menus:

- monitor command use
- reorder so most frequent on top ☺
- but order keeps changing ☹

Boxes
Fiona
Esther
Miriam
Colin
Tom
Devina
Gordon
Geoff
. . .

appropriate intelligence

- often simple heuristics
- combined with the right interaction

rules of standard AI interfaces

1. it should be right as often as possible
2. when it is right it should be good

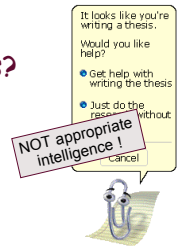
good for demos
look how clever it is!

rules of appropriate intelligence

1. it should be right as often as possible
2. when it is right it should be good
3. when it isn't right ... it shouldn't mess you up

} what makes a system really work!

Hit or a Miss?

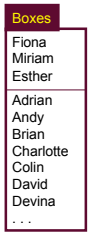


✗ paper clip
- can be good when it works
- but interrupts you if it is wrong

✓ Excel 'Σ' button
- guesses range to add up
- very simple rules (contiguous numbers above/to left)
- if it is wrong ... simply select what you would have anyway

... for menus

- small number of most popular at top quick when it gets it right
- alphabetic below still easy to scan when it isn't

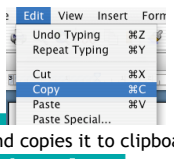


designing appropriate intelligence onCue

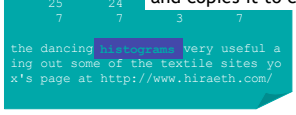
- intelligent toolbar
- sits at side of the screen
- watches clipboard for cut/copy
- suggests useful things to do with copied date

onCue in action

user selects text



and copies it to clipboard



slowly icons fade in

onCue appropriate?

1. it should be right as often as possible
   - uses simple heuristics:
     e.g. words with capitals = name/title
2. when it is right it should be good
   - suggests useful web/desktop resources
3. when it isn't right it shouldn't mess you up
   - slow fade-in means doesn't interrupt

✓