Data Mining and Applied Statistics

1. Introduction to statistics, probability theory, distributions
2. Statistical Inference, Central limit theorem, Confidence Interval, Hypothesis testing
3. Data Mining Process, ~~Preprocessing~~ Exploratory Data Analysis
4. Association Rule Mining
5. Clustering
6. Classification
7. Regression
8. Ensemble Learning, Preprocessing

Cloud Computing

1. Introduction
2. Technology & Types
3. Parallel Computing
4. Distributed System
5. Big Data
6. Map Reduce & Hadoop
7. Map Reduce Design Patterns
8. DM, LA, IR in the cloud

~~Cloud Computing~~ INDEX - II

V Information Retrieval & Text Mining

1. Boolean Retrieval
2. Term Vocabulary & Posting lists
3. Dictionary & Tolerant Retrieval
4. Index construction & Compression
5. Scoring, Term-weights, Vectorspace Model
6. Evaluation, Relevance, Query Expansion
7. Text ~~Mining~~ Clustering & Classification
8. Recommender Systems
9. Social Network Analysis
10. Opinion Mining
11. Information Extraction.

VI Semantic Web Technologies

1. Introduction
2. RDF - I
3. RDF - II
4. OWL - I
5. OWL - II & upper layers
6. networks, webarchitecture, HTTP
7. web technologies
8. web basics, crawling 2.0

2-phase locking protocol
- read, write locks
- strong, strict 2-phase locking
  R, W, wait, rel.

- growing/release phase
- 2PL o/ps is serializable

→ Deadlocks
- wait-for graphs
- about one
- prevention
  - timestamp
  - priority
  - combination

4. Stored Procedures, Triggers, Active Databases

→ Stored procedures :- code shared in dbms
SQL/PSM (queries + updates)

→ Triggers :- (event based)
Calls procedures

→ Active Databases :-
- supports triggers
  - events
  - condition
  - action

- ins/del/update
- cascading
- statement/row level
- Applications
  - constraint violations
  - views
  - logging
  - stats, authorization

Parallel databases

- parallel processing tech + DBMS
  - shared mem
  - shared disk
  - shared nothing
- throughput / response time
- speed-up (faster proc.) / scale-up (more proc.)
- intra / inter query parallelism
  - round robin
  - hash
  - range
- data partitioning (sharding)
  - h
  - v
- parallel impl. of rel. ops.
  - split & merge parallel streams of data
    - sort
      - parallel merge sort
    - proj & sel
      - trivial
    - join
      - parallel nested loop join
      - parallel sort merge join
      - parallel hash join

6. ~~Distributed Databases~~ Database Design

→ DBD: data modeling → refinement (relns)
(ER)

Files ← SQL Tables
(physical) (conceptual)

→ conceptual schema: relational model + Functional dependencies

→ normalization: eliminate anomalies

→ ER → reln. tables

→ problematic DB design
- redundancy
- anomalies

soln: decomposition
- loss less-join, dependency preservation

how :- functional dependency, normal form

dimension hierarchies

operations: - histograms
- summarize at diff levels
roll-up drill-down
- pivoting
cross-tabulation

→ Data Cube
- n-dimensional aggregate
generate powerset of aggregating columns

→ Roll up / drill down - bottom up / top down

→ indexes: bitmap
join
views

→ decision support: window queries, top-k,
online aggregation

APPLIED STATISTICS

1. Introduction to statistics, probability theory and distributions

→ Data Lifecycle: Business understanding → Data Understand
↓
Data prep → Modeling → Eval → Deploy

→ statistics → descriptive
→ inferential (sample)

→ Population → sample → statistic
describe
infer

→ Data → Categorical → nominal, ordinal
→ Interval → discrete, continuous

→ Measures of location: mean, median, mode, skew.

→ measures of dispersion: range, variance, quartile

- linear regression
  - target variable 'y' linearly dependent on attributes

- isotonic regression
  - non-linear, correct pairs of points, piecewise linear

- SVM for regression

- local regression - kNN + linear regression

- decision tree + regression
  - reg tree (piecewise constant fn)
  - model tree (piecewise linear fn)

- use ANN for regression

Ensemble

9. ~~Online~~ Learning, ~~Time series~~
   - wisdom of crowds
   - combining predictions from diff learners
   - accuracy, diversity
   - voting
     - class (most voted)
     - reg (avg)
   - bagging
     - diff samples for diff learners
     - learn models & combine
     - randomization (randomize the learning alg instead of input data)
       - eg - random forests
     - weighted voting with confidence values
   - boosting
     - set of classifiers one after another (later classifiers focus on errors by earl
     - iterations
   - stacking
     - learn a function to combine predictions

- model-based
  - item-based
    (similarity betn items)
    (cosine sim)
  - ARM
  - prob methods
    - bayes theorem

- content based reco
  - utilize item information
  - keyword overlap
  - content rep / user-profile
  - knn

- hybrid
  - voting, weights

- evaluating
  - accuracy
  - mean squared error
  - precision, recall, f-1

9. Social Network Analysis

→ source of data: web crawls, blogs, social n/ws, community data

→ networks:
- graph < directed / undirected
- vertex, arc, edge
- degree distribution
  - freq count
  - prob distn
  - avg degree
- graph density
  - L / L_{max}
- clustering coeff
  (density of connections in neighborhood)
- adjacency matrix, bipartite graph, paths,
- n/w diameter, avg distance

community detection

- set of frequently interacting actors
- connected graph, components, bridge, cut-vertex, giant component, isolates
- sce / wce (undir) (undir)
- web: central [illegible] subgraph
- k-cores: max undir subgraph in which each vertex has atleast degk
- cliques: max connected complete subgraph >3 vertices
- divisive hierarchical clustering (remove weakest edge)
- edge betweeness: no. of shortest paths that pass through
- islands
- prominence: centrality / prestige (inv) (target)
- power - law distributions (degree dist)
- small-world n/w (clustering coeff, shortest path length)

10. Sentiment Analysis & Opinion Mining

→ SA: classifying polarity of given text
→ polarity values: +ve, -ve, neutral

declared / feature level \

→ mine opinions in user generated content

regular / comparisons \

→ opinion, target of opinion, opinion holder
→ entity e : A (finite set of aspects)
= $(a_1, a_2 .. a_n)$
model of entity
→ model of review
- comments on subset of aspects
- choose a word to desc entity / an part
+, -, neutral

→ sensor filter
- checks whether extracted info is plausible
- utilizes discriminator phrases
- pointwise mutual info

→ bootstrapping
takes domain independent rule templates, predicates & learns extraction rules & discriminators

→ extractor
from www & extraction rules, produces extractions

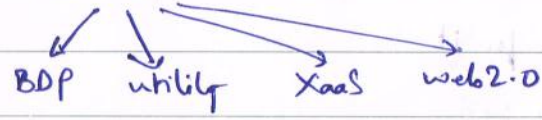
→ assessor: generates valid extractions

CLOUD COMPUTING

1. Introduction

supercomputer, grid → cloud

→ cloud



→ principles
- pooled resources: avail to all
- virtualization: high utilization of h/w (1 m/c multiple instances)
- elasticity: dynamic scale
- automation: no manual
- metered billing: payg

→ benefits
- economic
- agility
- efficiency
- security

web browser - client / server

servers - sshd, httpd: listening on port

multithreaded web server - modelled by servlets

URL: address of doc

html:- describe structure of doc

tags	anchor key
attributes	tables
lists	layout, image files
comments, special chars	

content v/s presentation

CSS - display html docs

6. Networking, html, ~~html~~, http, ~~web arch~~

→ protocol hierarchy
- layers, protocols, interfaces

→ n/w arch
- set of layers + protocols
  - layers → services → primitives
  - protocols → rules of convo

- OSI reference model
App - Pres - Session - Transport - N/w - DL - Phy

- sending data - connectionless / connection oriented
(pkt) (ckt)

- TCP/IP
  - connection oriented
  - UDP: connectionless
  - lower level

- IP v4, v6 addresses
classes A, B, C, D, E ; CIDR

→ adapter
- resolve incompatible interfaces

→ visitor
- define new op without changing class

→ singleton
- only 1 instance of a class created

→ proxy
- for an object to control references to it

→ command
- encapsulate request to it

→ factory
- Create objects without exposing instantiation logic to clients

6. Typing, Memory management

→ Typed v/s untyped languages
- for every operation, types of data for which it is applicable

- weak / strong typing
↓
type-safe
prevent value of one type to be treated as another

- static / dynamic typing
↓ compile time
↘ run-time (no-type for variable during prg.)

→ Activation records : allocated when proc is entered & dealloc when proc is exited
(info needed by single exec of proc)

→ software requirement analysis & specification

- customer based
- requirements gathering
  - functional
  - non-functional
- requirements analysis
  - identify inconsistency, defects etc
  - data flow diagram
  - entity rel diagram
  - data dictionary
- requirements documentation
  - SRS: software req. spec
  - functionality
  - external interfaces
  - performance
  - design constraint

→ software design

- design: given SRS doc, propose a solution

- modularity:
  - coupling - degree of independence betn modules (inter dependence)
    - low (better)
    - high
  - coupling
    - data
    - stamp
    - control
    - external
    - common
    - content
  - cohesion: functionally related 2 modules are

→ software project planning

- size estimation
  - LOC
  - function count
- overall cost estimation
- risk estimation

- reusability
- scalibility
- complexity
- cost
- open-standard protocol stack

- Technical details
  - XML data transfer
  - SOAP: HTTP + XML
  - client + UDDI registry + endpoint
  - WSDL
  - UDDI
  - component model
    - interface
    - implementation

FUNDAMENTALS OF COMPUTING

1. Digital logic, arithmetic

→ computers binary: simple, easy to build, more efficient

→ logic: valid inferencing & correct reasoning

→ stmt is T/F

connectives: $\wedge, \neg, \vee, \rightarrow$

→ truth tables

→ boolean logic in computers

bits 0, 1

logic gates:- AND, OR, NOT

→ boolean exp $\Leftrightarrow$ boolean ckt

→ more gates: XOR, NAND, NOR

→ 2^{2^n} boolean function

→ $\{\neg, \wedge, \vee\}$ is universal set of boolean fns

→ NAND, NOR are also universal

→ file Mgt
- 2° storage
- sys utility progs
- identifying locate
- quick random access, ease of update, economy of storage, reliable
- directory — tree
- path name
  - absolute
  - relative
- file sharing
  - ACL / concurrency
- space allocation
  - index
  - contiguous
  - continuous

ry, except, else, finally
methods, attrs special
terator, generator → iter with a for (yield)

PYTHON

→ interpreted
→ dynamically typed
→ object system, iterator, generators, exceptions, functions as independent entities, types, braces for indentation

→ Types
- tuples (immutable) (fast) — -ve index (from the end)
- lists (mutable) (slow)
- strings (immutable)

→ extend, + (new object) (inplace)

→ Dictionaries
→ maps (key-value)

→ boolean exp
→ if, for, break, continue
→ abuse of range() [anti-pattern]
→ list comprehensions
- gen a list by applying function to each member

→ functions : no overloading, lambda, first class objects
→ inheritance, import
→ strings :- % formatting