# COMP9318 Review

Wei Wang @ UNSW

May 21, 2017

# Course Logisitics

- **THE** formula:

$$
\begin{aligned}
mark &= 0.55 \cdot exam + 0.15 \cdot (ass1 + proj1 + lab) \\
mark &= \textbf{FL}, \text{ if } exam < 40 \\
lab &= avg(best\_of\_3(lab1, lab2, lab3, lab4, lab5))
\end{aligned}
$$

- proj1 and ass1 will be marked ASAP; we aim at delivering the result before the exam
- Pre-exam consultations:
  - 12 Jun: 1300–1500, K17-508
  - 13 Jun: 1300–1500, K17-508
- Course feedback: via comments in the course survey or private messages to me on the forum. We are particularly interested in aspects such as **coverage**, **difficulty levels**, **use of python/Jupyter**, **project**, and **background required**.

## Note

(1) The final exam mark is important and you must achieve at least 40!
(2) Supplementary exam is only for those who cannot attend final exam.

# About the Final Exam

- **Time**: 0900 – 1100, 14 Jun 2016 (Wed), 10 minutes reading time + 2 hr closed-book exam.
- **Accessories**: *UNSW Approved Calculator*. Note: watches are prohibited.
- Designed to test your *understanding* and familiarity of the core contents of the course.
- Answer 1 + 6 questions out of 9 questions.
  - Q1: short answer (can use your own words) and **compulsory**.
  - Choose 6 from Q2 to Q9; thers will requires some “calculation” (i.e., similar to tute/ass questions)

# About the Final Exam /2

- Read the instructions carefully.
- Use your time wisely. Don't spend too much time if stuck on one question or writing excessively long answers on Q1.

## Tips

(1) Write down intermediate steps. (2) Know how to do $\log_2(x)$ on your calculator. (3) Work on "easy" questions first (but start the answer on a new page on the booklet).

## Disclaimer

*We will go through the main contents of each lecture. However, note that it is by no means exhaustive.*

# Introduction

- DM vs. KDD
- Steps of KDD; iterative in nature; results need to be validated.
- Database (efficiency) vs. Machine learning (effectiveness) vs. Statistics (validity):
- Able to cast a real problem into a data mining problem.

# Data Warehousing and OLAP

- Understand the four characteristics of DW (DW vs. Data Mart)
- Differences between OLTP and OLAP
- Multidimensional data model; data cube;
  - fact, dimension, measure, hierarchies
  - cuboid, cube lattice
  - three types of schemas
  - four typical OLAP operations
  - ROLAP/MOLAP/HOLAP
- Query processing methods for OLAP servers, including the BUC cubing algorithm.

**NOT** needed:

- Design good DW schemas and perform ETL from operational data sources to the DW tables.

# Data Preprocessing

- Understand that real data is “dirty” (incomplete, noisy, inconsistent)
- How to handle missing data?
- How to normalize the data?
- How to handle noisy data? different binning/histogram method (including V-optimal and MaxDiff)
- How to discretize data?

**NOT** needed:

- Feature selection and reduction (e.g., PCA, Random Projection, t-SNE)

# Classification and Prediction

- Classification basics:
  - overfitting/underfitting; cross-validation
  - Classification vs prediction; vs clustering (unsupervised learning); eager learning vs. lazy learning (instance-based learning)
- Decision tree:
  - The ID3 algorithm
  - Decision tree pruning
  - Derive rules from the decision tree
  - The CART algorithm (with gini index)
- Naive Bayes classifier
  - Smoothing
  - Two ways to apply NB on text data
- Logistic regression/MaxEnt classifier; Maximum likelihood estimation of the model parameters + regularization; Gradient ascend.

# Cluster Analysis

- Clustering criteria: minimize intra-cluster distance + maximize inter-cluster distance
- Distance/similarity
    - how to deal with different types of variables
    - distance functions: $L_p$
    - metric distance functions

# Cluster Analysis /2

- Partition-based Clustering: *k*-Means (algorithm, advantages, disadvantages, . . . )
- Hierarchical Clustering: agglomerative, single-link / complete-link / group average hierarchical clustering

# Association Rule Mining

- Concepts:
  - Input: transaction db
  - Output: (1) *frequent* itemset (via *minsup*); (2) association rules (via *minconf*)
- Apriori algorithm:
  - *Apriori property* (2 versions)
  - The Apriori algorithm
    - How to find frequent itemsets?
    - How to derive the association rules?

# Association Rule Mining /2

- FP-growth algorithm:
  - How to mine the association rule using FP-trees?
- Derive association rules from the frequent itemsets.

# Recommendation System

- Problem definition and its input.
- Three major approaches, including their pros and cons.
  - Content-based
  - Collaborative filtering; pearson correlation
  - Matrix factorization-based; stochastic gradient descend.
- Variants of Matrix factorization.

**NOT** needed:

- The detailed model used by the Netflix competition winner.

Thanks You and Good Luck!