Introduction

LING 572

Fei Xia

Week 1: 1/05/2010

Outline

- General course information
- Course contents

General course information

Prerequisites

- CS 326 (Data Structures) or equivalent:
  - Ex: hash table, array, tree, …
- Stat 391 (Prob. and Stats for CS) or equivalent: Basic concepts in probability and statistics
  - Ex: random variables, chain rule, Bayes' rule
- Programming in C/C++, Java, Perl, Python, or Ruby
- Basic unix/linux commands (e.g., ls, cd, ln, sort, head): tutorials on unix
- LING570: if you did not take it with me, you need to go over the slides and assignments for my 570 from last quarter.
- **If you don't meet all the prerequisites, you need to email me by 6pm tomorrow.**

Topics covered in Ling570

- LM, ngram, and smoothing
- HMM and POS tagging
- Classification task and Mallet
- Chunking, NE tagging, clustering

Grades for LING572

- No midterm or final exams.
- Programming Assignments (9): 90%
- Reading assignments (4-5): 10%
- Class participation: 10%
  - 50%: ask questions in class and on GoPost
  - 50%: help others on GoPost or in/after class

- Remove the lowest score to calculate average.
- The average is then mapped to the final grade.

Tentative mapping from the class average to the final grade

98-100	4.0	77-79	3.3
95-97	3.9	74-76	3.2
92-94	3.8	71-73	3.1
89-91	3.7	68-70	3.0
86-88	3.6	65-67	2.9
83-85	3.5	62-64	2.8
80-82	3.4	59-61	2.7

Office hours

- Fei:
  - Email:
    - Email address: fxia@uw.edu
    - Subject line should include "**ling572**"
    - The 36-hour rule: it works both ways
  - Office hour:
    - Time: Thurs 11am-noon ??
    - Location: Padelford A-210G

TA hours

- Ryan Georgi
  - Email: rgeorgi@uw.edu
  - Time:
    - M, T, W, Th: 3-4pm
    - F: 1-2pm
  - Location: treehouse??

Questions about grades

- If you have any questions about hw grades, please email Ryan first.

- For any remaining issues, email me and cc Ryan.

Slides

- The slides will be online before class.
- The final version will be uploaded a few hours after class.
- “Additional slides” are not required and not covered in class.

Url, GoPost, Email

- Course url: **http://courses.washington.edu/ling572**
  - Syllabus (incl. slides, assignments, and papers):
  - GoPost:
  - CollectIt:

- GoPost: Most course-related questions should go to GoPost, including the urls of recordings.

- Email: you should use it ONLY for confidential subjects.

- Please check your emails and GoPost at least once per day.

GoPost

- GoPost is mainly a venue for student discussion.

- I am NOT going to answer all the questions:
  - Some questions have been answered already.
  - As for others, I prefer that students would work out the answers by themselves.

GoPost (cont)

- Main discussion areas:
  - Announcements
  - General information
  - Recordings
  - Grades
  - Hw1, Hw2, …
- A discussion area can have multiple threads, and each thread can have multiple posts.
- Start a new thread when the subject changes.
- Each thread should have a clear title: e.g., “Q1: …”

GoPost (cont)

- Posts on GoPost do not change hw, so you should be able to complete hw without relying on GoPost.
- Going through posts can be time consuming, and some posts could be misinterpreted if you are not “there”.
- You need to decide what’s the best way to take advantage of GoPost.

Reading assignments

- You will answer some questions about the papers that will be discussed in next class.
- The questions are on teaching slides, and there are no separate documents for them.
- Your answers should be concise and no more than a few lines.
- Your answers are due before the next class. Bring the hardcopy of your answers to class.

Programming assignments

- Due date: every Thurs at 11:55pm unless specified otherwise.
- The submission area is closed **4** days after the due date.
- There is 1% penalty for every hour after the due date.

Programming assignments

- Programming languages: C, C++, Java, Perl, or Python
- Write a simple shell script
- Follow the instructions in the assignments, including
  - command line format:
    cat input | foo.sh arg1 arg2 … > output
  - file format
  - the probability model
  - Naming convention: hw1.notes
- Your code must run on Patas

Shell script

- An example: output the first n lines in STDIN
  - All under dropbox/08-09/572/code/code-samples/

- Write your code:

  Perl: cat ex | ncat.pl 5 > t1 2>t2

  Python: cat ex | ncat.py 5 > t1 2>t2

- Use a shell script: ncat.sh
  - cat ex | ncat.sh 5 1>t1 2>t2

Shell script (cont)

```
#!/bin/sh

./ncat.pl $@                    # Perl

./ncat.py $@                    # Python

./ncat $@                       # C
```

➔ See ~/dropbox/08-09/572/code/code-samples/

Homework Submission

- Use “Collect it”: submit the tar file.
  - E.g., tar –cvf hw1.tar hw1_dir

- Each submission includes
  - a note file: hw1.(txt|doc|pdf) for hw1.
    - If your code does not work, explain in the note file what you have implemented so far.
  - a set of shell scripts: e.g., kNN.sh
  - source code: e.g., kNN.C
  - binary code (for C/C++/Java): kNN.out
  - data files if any.
  - The TA will NOT compile or debug your code.

Patas

- If you need to have a patas account, you need to email linghelp@u.washington.edu right away to get an account.

- The directory for LING572:

  ~/dropbox/09-10/572/

  – hw1/, hw2/, ….: Assignments and solution

  – misc_slides/: Solution to exams and misc slides that are not on the course url.

- For jobs that run more than 5 minutes, use the cluster submission commands: see slides from 1/14

Summary of assignments

	Assignments (hw)	Reading assignments
Num	9	5-6
Distribution	Download from the course url	
Discussion	Allowed	
Submission	Collect It	Bring to class Not graded
Due date	11:55pm every Thurs	Before next class
Extension	1% penalty per hour	Disallowed
Estimate of hours	10-30 hours	2-6 hours
Solution files	On Patas	Discussed in class

Workload

- On average, students will spend around
  - 20 hours on each assignment
  - 3 hours on lecture time
  - 2-3 hours on GoPost
  - 2-3 hours on each reading assignment
  - ➔ 25-30 hours per week

- You need to be realistic about how much time you have for 572.

- I will have a thread on “time spent” for each assignment on GoPost. I will appreciate it if you could reply to that post.

Programming assignments

- Try to reuse code from previous assignments.

- Results:
  - No need to get exactly the same results: if the gold standard is 83.8, getting 83.1 is fine.

  ➔ spend time on high-level ideas, not on debugging.

- Teamwork: (??)
  - Discuss pseudo code together, but only one person has to type in the code and debug

Extension and incomplete

- Extension and incomplete are given only under extremely unusual circumstances (e.g., health issues, family emergency).

- The following are NOT acceptable reasons for extension:
  - My code does not quite work.
  - I have a deadline at work.
  - I am going to be out of town for a few days.
  - …

Course contents

Types of ML problems

- Classification problem
- Estimation problem
- Clustering
- Discovery
- …

➔A learning method can be applied to one or more types of ML problems.

➔We will focus on the classification problem.

Course objectives

- Covering basic statistical methods that produce state-of-the-art results
- Focusing on classification and sequence labeling problems
- Some ML algorithms are complex. We will focus on basic ideas, not theoretical proofs.

Main units

- Simple classification algorithms (2 weeks)
  - kNN
  - Decision tree
  - Naïve Bayes
- Advanced classification algorithms (4 weeks)
  - MaxEnt
  - CRF
  - SVM

Main units (cont)

- Sequence labeling algorithms and SSL (1.5 weeks)
  - TBL
  - EM (if time permits)
  - Introduction to semi-supervised learning

- Misc topics (2.5 weeks)
  - Introduction
  - Two packages: Mallet and libSVM
  - Feature selection
  - Converting Multi-class to binary classification problem
  - Review and summary

Questions for each ML method

- Six methods:
  - kNN and SVM
  - DT and TBL
  - NB and MaxEnt

- Modeling:
  - what is the model?
  - What kind of assumption is made by the model?
  - How many types of model parameters?
  - How many “internal” (or non-model) parameters?
  - …

Questions for each method (cont)

- Training: how to estimate parameters?
- Decoding: how to find the “best” solution?
- Weaknesses and strengths:
  - Is the algorithm
    - robust? (e.g., handling outliners)
    - scalable?
    - prone to overfitting?
    - efficient in training time? Test time?
  - How much data is needed?
    - Labeled data
    - Unlabeled data

Coming up

- If you have any question about the course, email me by 9am tomorrow.

- No class on 1/7, due to LSA at Baltimore. The lecture is recorded and the urls are at GoPost.
  - Information theory
  - Probability
  - Classification task (from ling570)
  - Mallet (from ling570)

- Hw1 is due on 1/14.