Introduction

LING 572
January 8, 2019
Gina-Anne Levow

Outline

- General course information

- Course contents

General course information

Course web page

- Canvas page: https://canvas.uw.edu/courses/1257221
- Navigation menu:
  - Home: Office hours, links to zoom room, schedule, and course policy.
  - Announcements: please check at least daily
  - Syllabus:
    - Prerequisites, textbooks, link to schedule
    - Course summary: generated automatically
  - Discussions
  - Assignments: assignment and submission
  - Grades
  - Conferences: for remote office hours
  - People: you can form study groups and get workspace for the group

Office hours

- Gina:
  - Email: levow@uw.edu

  - Office hours:
    - In-person: 2:30-3:30pm on Thurs (GUG 418B)
    - remote: 2-3pm on Friday via Canvas

TA office hours

- David Inman:
  - Email: davinman@uw.edu
  - Office hours:
    - In-person: Tues 3-4pm: GUG TA space
    - Remote: Mon 9-10am via Canvas

Online Option

- The link to Zoom is on the home page: https://washington.zoom.us/j/842108296
- Please enter meeting room 5 mins before start of class
  - Try to stay online throughout class
  - Please mute your microphone
  - Please use the chat window for questions

Communication

- If you prefer, you can use your Canvas inbox for all course-related emails:
  - Easy to find people’s email addresses
  - Emails can be grouped by courses.
- If you don’t use Canvas to send email, please include ling572 in your subject line of email to us.
- If you do not check Canvas often, please remember to set Account: Notifications in Canvas: e.g., “Notify me right away”, “send daily summary”.
- Do not send email to the whole class except for emergency.
- For a non-urgent question, post to discussion board or ask in class / during office hours.
- We will use Canvas:Announcement for important messages and reminders.

Programming assignments

- Due date: every Weds at 11pm unless specified otherwise.

- The submission area closes two days after the due date.

- Late penalty:
  - 1% for the 1st hour
  - 10% for the 1st 24 hours
  - 20% for the 1st 48 hours

Programming languages

- Recommended languages:
  - C/C++/C#, Java, Python, Perl, Ruby, Mono, Jython
  - If you want to use a non-default version, use the correct path in your script.
  - See dropbox/18-19/572/languages
- If you want to choose a language that is NOT on that list:
  - You should contact Gina about this ASAP.
  - If the language is not currently supported on patas, it may take time to get that installed.
  - If your code does not run successfully, it could be hard for the grader to give partial credit for a language that (s)he is not familiar with.
- Your code must run, and will be tested, on patas.

Homework Submission

- For each assignment, submit two files through Canvas:
  - A note file: readme.txt or readme.pdf
  - A gzipped tar file that includes everything: hw.tar.gz (not hwX.tar.gz)

```
      cd hwX/          # suppose hwX is your dir that includes all the files
      tar -czvf hw.tar.gz *
```

- Before submitting, run check_hwX.sh to check the tar file: e.g.,

```
     /dropbox/18-19/572/hw1/check_hw1.sh hw.tar.gz
```

- check_hwX.sh checks only the existence of files, not the format or content of the files.

- For each shell script submitted, you also need to submit the source code and binary code: see 572/hwX/submit-file-list and 572/languages

Rubric

- Standard portion: 25 points
  - 2 points: hw.tar.gz submitted
  - 2 points: readme.[txt|pdf] submitted
  - 6 points: all files and folders are present in the expected locations
  - 10 points: program runs to completion
  - 5 points: output of program on patas matches submitted output
- Assignment-specific portion: 75 points

Regrading requests

- You can request regrading for:
  - wrong submission or missing files: show the timestamp
  - crashed code that can be easily fixed (e.g., wrong version of compiler)
  - output files that are not produced on patas
- At most two requests for the course.
- 10% penalty for the part that is being regraded.
- For regrading and any other grade-related issues: you must contact the TA within a week after the grade is posted.

Reading assignments

- You will answer some questions about the papers that will be discussed in an upcoming class.
- Your answer to each question should be concise and no more than a few lines.
- Your answers are due at 11am. Submit to Canvas before class.
- If you make an effort to answer those questions, you will get full credit.

Summary of assignments

	Assignments (hw)	**Reading assignments**
Num	9 or 10	4 or 5
Distribution	Canvas and patas	Canvas
Discussion	Allowed	
Submission	Canvas	
Due date	11pm every Weds	11am on Tues or Thurs
Late penalty	1%, 10%, 20%	No late submission accepted
Estimate of hours	10-15 hours	2-4 hours
Grading	Graded according to the rubrics	Checked

Workload

- On average, students will spend around
  - 10-20 hours on each assignment
  - 3 hours on lecture time
  - 2 hours on Discussions
  - 2-3 hours on each reading assignment

   15-25 hours per week; about 20 hrs/week

- You need to be realistic about how much time you have for 572. If you cannot spend that amount of time on 572, you should take 572 later when you can.
- If you often spend more than 25 hours per week on 572, please let me know. We can discuss what can be done to reduce time.

Extensions and incompletes

- Extensions and incompletes 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.
  - …

Final grade

- Grade:
  - Assignments: 100% (lowest score is removed)
    - All the reading assignments are treated as one “regular” assignment w.r.t. “the lowest score”.
  - Bonus for participation: up to 2%
  - The percentage is then mapped to final grade.
- No midterm or final exams
- Grades in Canvas:Grades
- TA feedback returned through Canvas:Assignments

Course Content

Prerequisites

- CSE 373 (Data Structures) or equivalent:
  - Ex: hash table, array, tree, …
- Math/Stat 394 (Probability I) or equivalent: Basic concepts in probability and statistics
  - Ex: random variables, chain rule, Bayes' rule
- Programming in C/C++, Java, Python, Perl, or Ruby
- Basic unix/linux commands (e.g., ls, cd, ln, sort, head): tutorials on unix
- LING570
- **If you don't meet the prerequisites, you should wait and take ling572 later.**

Topics covered in Ling570

- FSA, FST
- LM and smoothing
- HMM and POS tagging
- Classification tasks and Mallet
- Chunking, NE tagging
- Information extraction
- Word embedding and NN basics

Textbook

- No textbook
- Readings are linked from the course website.
- Reference / Background:
  - Jurafsky and Martin, *Speech and Language Processing: An Introduction to NLP, CL, and Speech Recognition,* 2nd edition, 2008.
  - Manning and Schutze, *Foundations of Statistical NLP*

Types of ML problems

- Classification problem
- Regression 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 many statistical methods that are commonly used in the NLP community

- Focusing on classification and sequence labeling problems

- Some ML algorithms are complex. We will focus on basic ideas, not theoretical proofs.

Main units

- Basic classification algorithms (1.5 weeks)
  - kNN
  - Decision trees
  - Naïve Bayes

- Advanced classification algorithms (5-6 weeks)
  - MaxEnt
  - CRF
  - SVM
  - Introduction to neural networks

Main units (cont)

- Other learning methods (1 week)
  - TBL
  - Introduction to semi-supervised learning (??)

- Misc topics (1-2 weeks)
  - Introduction
  - Feature selection
  - Converting Multi-class to binary classification problem
  - Review and summary

Questions for each ML method

- Learning methods:
  - kNN and SVM
  - DT and TBL
  - NB and MaxEnt
  - Perceptron and NN

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

Questions for each method (cont’d)

- Training: how can we estimate parameters?

- Decoding: how can we find the “best” solution?

- Weaknesses and strengths:
  - Is the algorithm
    - robust? (e.g., handling outliers)
    - scalable?
    - prone to overfitting?
    - efficient in training time? Test time?
  - How much data is needed?
    - Labeled data
    - Unlabeled data

Please go over self-study slides

- All are on the ling572 website.

- All have been covered in Ling570
  - Probability Theory
  - Overview of Classification Task
  - Using Mallet
  - Patas and Condor
  - Word embedding for prior quarter’s ling570