Introduction + Information Theory

LING 572
January 7, 2020
Shane Steinert-Threlkeld

Adapted from F. Xia, ‘17

Outline

- Background
- General course information
- Course contents
- Information Theory

Early NLP

- Early approaches to Natural Language Processing
  - Similar to classic approaches to Artificial Intelligence

Early NLP

- Early approaches to Natural Language Processing
  - Similar to classic approaches to Artificial Intelligence
  - Reasoning, knowledge-intensive approaches

Early NLP

- Early approaches to Natural Language Processing
  - Similar to classic approaches to Artificial Intelligence
  - Reasoning, knowledge-intensive approaches
  - Largely manually constructed rule-based systems

Early NLP

- Early approaches to Natural Language Processing
  - Similar to classic approaches to Artificial Intelligence
  - Reasoning, knowledge-intensive approaches
  - Largely manually constructed rule-based systems
  - Typically focused on specific, narrow domains

Early NLP: Issues

- Rule-based systems:

Early NLP: Issues

- Rule-based systems:
  - Too narrow and brittle
    - Couldn’t handle new domains: out of domain → crash

Early NLP: Issues

- Rule-based systems:
  - Too narrow and brittle
    - Couldn't handle new domains: out of domain → crash
  - Hard to maintain and extend
    - Large manual rule bases incorporate complex interactions
    - Don't scale

Early NLP: Issues

- Rule-based systems:
  - Too narrow and brittle
    - Couldn't handle new domains: out of domain → crash
  - Hard to maintain and extend
    - Large manual rule bases incorporate complex interactions
    - Don't scale
  - Slow

Reports of the Death of NLP…

- ALPAC Report: 1966
  - Automatic Language Processing Advisory Committee

Reports of the Death of NLP…

- ALPAC Report: 1966
  - Automatic Language Processing Advisory Committee
  - Failed systems efforts, esp. MT, lead to defunding

Reports of the Death of NLP…

- ALPAC Report: 1966
  - Automatic Language Processing Advisory Committee
  - Failed systems efforts, esp. MT, lead to defunding

  - Example: (Probably apocryphal)
    - English → Russian → English MT
    - “The spirit is willing but the flesh is weak.”→
    - “The vodka is good but the meat is rotten.”

…Were Greatly Exaggerated

- Today:
- Alexa, Siri, etc converse and answer questions
- Search and translation
- Watson wins Jeopardy!

So What Happened?

- Statistical approaches and machine learning

So What Happened?

- Statistical approaches and machine learning
  - Hidden Markov Models boosted speech recognition

So What Happened?

- Statistical approaches and machine learning
  - Hidden Markov Models boosted speech recognition

  - Noisy channel model gave statistical MT

So What Happened?

- Statistical approaches and machine learning
  - Hidden Markov Models boosted speech recognition
  - Noisy channel model gave statistical MT
  - Unsupervised topic modeling
  - Neural network models, esp. end-to-end systems and (now) pre-training

So What Happened?

- Many stochastic approaches developed 80s-90s
- Rise of machine learning accelerated 2000-present
- Why?

So What Happened?

- Many stochastic approaches developed 80s-90s
- Rise of machine learning accelerated 2000-present
- Why?
  - Large scale data resources
    - Web data (Wikipedia, etc)
    - Training corpora: Treebank, TimeML, Discourse treebank

So What Happened?

- Many stochastic approaches developed 80s-90s
- Rise of machine learning accelerated 2000-present
- Why?
  - Large scale data resources
    - Web data (Wikipedia, etc)
    - Training corpora: Treebank, TimeML, Discourse treebank
  - Large scale computing resources
    - Processors, storage, memory: local and cloud

So What Happened?

- Many stochastic approaches developed 80s-90s
- Rise of machine learning accelerated 2000-present
- Why?
  - Large scale data resources
    - Web data (Wikipedia, etc)
    - Training corpora: Treebank, TimeML, Discourse treebank
  - Large scale computing resources
    - Processors, storage, memory: local and cloud
  - Improved learning algorithms (supervised, [un-/semi-]supervised, structured, …)

General course information

Course web page

- Course page: https://www.shane.st/teaching/572/win20/index.html
- Canvas: https://canvas.uw.edu/courses/1356316
  - Lecture recording
  - Assignment submission / grading
  - Discussion!

Communication

- Contacting teaching staff:
  - If you prefer, you can use your Canvas inbox for all course-related emails:
  - If you do send email, please include LING572 in your subject line of email to us.
  - We will respond within 24 hours, but only during “business hours” during the week.
- If you do not check Canvas often, please remember to set Account: Notifications in Canvas: e.g., “Notify me right away”, “send daily summary”.
- Canvas discussions:
  - All content and logistics questions
  - If you have the question, someone else does too. Someone else besides the teaching staff might also have the answer.
- We will use Canvas:Announcement for important messages and reminders.

Office hours

- Shane:
  - Email: shanest@uw.edu

  - Office hours:
    - Tuesday 2:30-4:30pm (GUG 418D + Zoom)

TA office hours

- Yuanhe Tian:
  - Email: yhtian@uw.edu
  - Office hours:
    - GUG 417 (the Treehouse)
    - Wed 3-4pm
    - Friday 10-11am

Online Option

- The link to Zoom is on the home page: https://washington.zoom.us/my/clingzoom
- 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

Programming assignments

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

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

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

Programming languages

- Recommended languages:
  - Python, Java, C/C++/C#
  - If you want to use a non-default version, use the correct/full path in your script.
  - See dropbox/19-20/572/languages
- If you want to choose a language that is NOT on that list:
  - You should contact Shane 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/19-20/572/hw2/check_hw2.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	Web and patas	Web
Discussion	Allowed	
Submission	Canvas	
Due date	11pm every Thurs	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 have exams / work in my other courses.
  - 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 single 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*
  - 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 [multinomial logistic regression]
  - CRF
  - SVM
  - Neural networks

Main units (cont)

- 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
  - NB and MaxEnt
  - 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 (hyperparameters)?
  - …

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 (and what kind of) data is needed?
    - Labeled data
    - Unlabeled data

Please go over self-study slides

- All are on the LING 572 website.

- All have been covered in Ling570
  - Probability Theory
  - Overview of Classification Task
  - Using Mallet
  - Patas and Condor [under Course Resources]

Information Theory

Information theory

- Reading: M&S 2.2, Cover and Thomas ch. 2

- The use of probability theory to quantify and measure “information”.

- Basic concepts:
  - Entropy
  - Cross entropy and relative entropy
  - Joint entropy and conditional entropy
  - Entropy of a language and perplexity
  - Mutual information

Entropy

- Intuitively: how ‘surprising’ a distribution is
  - high entropy = uniform; low entropy = peaked
- Can be used as a measure of
  - Match of model to data
  - How predictive an n-gram model is of next word
  - Comparison between two models
  - Difficulty of a speech recognition task

Entropy

- Information theoretic measure
- Measures information in model
- Conceptually, lower bound on # bits to encode

Entropy

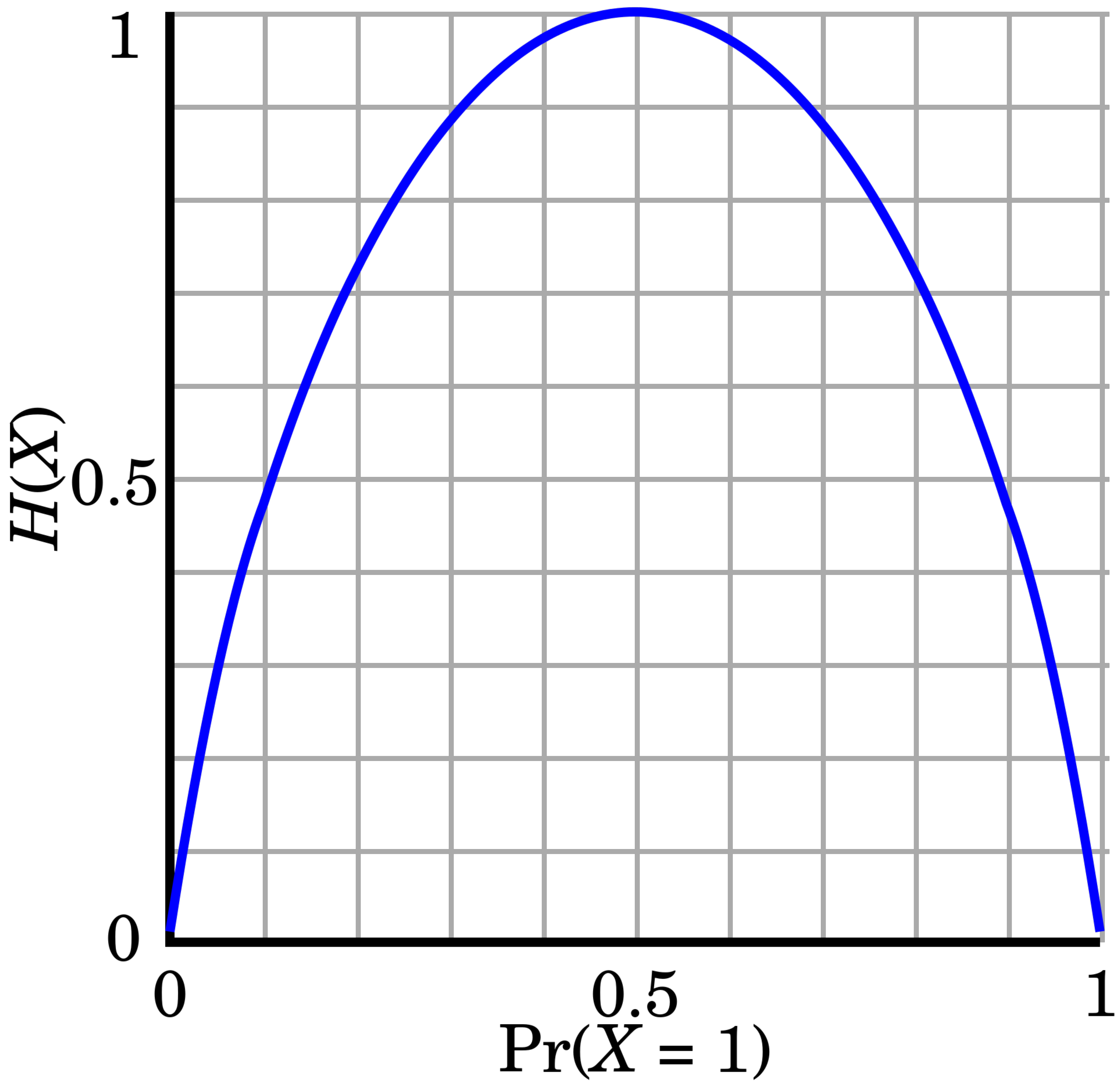
- Entropy is a measure of the uncertainty associated with a distribution.

$$H(X) = -\sum_x p(x) \log p(x)$$

Here, X is a random variable, x is a possible outcome of X.

- The lower bound on the number of bits that it takes to transmit messages.
- Length of the average message of an optimal coding scheme

Example 1: a coin-flip



Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i)

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, $p(i) = 1/8$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} p(i) \log p(i)$$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} 1/8 \log 1/8$$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} 1/8 \log 1/8 = -\log 1/8$$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} 1/8 \log 1/8 = -\log 1/8 = 3 \text{ bits}$$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} 1/8 \log 1/8 = -\log 1/8 = 3 \text{ bits}$$

  - Some horses more likely:
    - 1: ½; 2: ¼; 3: 1/8; 4: 1/16; 5-8: 1/64

$$H(X) = -\sum_{i=1}^{8} p(i) \log p(i)$$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} 1/8 \log 1/8 = -\log 1/8 = 3 \text{ bits}$$

  - Some horses more likely:
    - 1: ½; 2: ¼; 3: 1/8; 4: 1/16; 5-8: 1/64

$$H(X) = -\sum_{i=1}^{8} p(i) \log p(i) = 1/2 \log 1/2 + 1/4 \log 1/4 + 1/8 \log 1/8 + 1/16 \log 1/16 + 4/64 \log 1/64$$

Computing Entropy

- Picking horses (Cover and Thomas)
- Send message: identify horse - 1 of 8
  - If all horses equally likely, p(i) = 1/8

$$H(X) = -\sum_{i=1}^{8} 1/8 \log 1/8 = -\log 1/8 = 3 \text{ bits}$$

  - Some horses more likely:
    - 1: ½; 2: ¼; 3: 1/8; 4: 1/16; 5-8: 1/64

$$H(X) = -\sum_{i=1}^{8} p(i) \log p(i) = 2 \text{ bits}$$

    - 0, 10, 110, 1110, 111100, 111101, 111110, and 111111.

➔ Uniform distribution has a higher entropy.

➔MaxEnt: make the distribution as “uniform” as possible.

Entropy = Expected Surprisal

$$H(X) = -\sum_x p(x)\log p(x) = \mathbb{E}_p - \log p(X)$$

Cross Entropy

- Entropy: $$H(X) = -\sum_x p(x)\log p(x)$$

- Cross Entropy: $$H_c(X) = -\sum_x p(x)\log q(x)$$

Here, p(x) is the true probability;

q(x) is our estimate of p(x).

$$H_c(X) \geq H(X)$$

Relative Entropy

- Also called Kullback-Leibler divergence:

$$KL(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)} = H_c(X) - H(X)$$

- A “distance” measure between probability functions p and q; the closer p(x) and q(x) are, the smaller the relative entropy is.

- KL divergence is asymmetric, so it is not a proper distance metric:

$$KL(p\|q) \neq KL(q\|p)$$

Joint and conditional entropy

- Joint entropy:

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(x,y)$$

- Conditional entropy:

$$H(Y|X) = \sum_x p(x) H(Y|X=x) = H(X,Y) - H(X)$$

Entropy of a language (per-word entropy)

- The cross entropy of a language L by model m:

$$H(L, m) = -\lim_{n\to\infty} \frac{\sum_{x_{1n}} p(x_{1n}) \log m(x_{1n})}{n}$$

- If we make certain assumptions that the language is “nice”*, then the entropy can be calculated as: (Shannon-Breiman-Mcmillan Theorem)

$$H(L, m) = -\lim_{n\to\infty} \frac{\log m(x_{1n})}{n} \approx -\frac{\log m(x_{1n})}{n}$$

Per-word entropy (cont'd)

- $m(x_{1n})$ often specified by a language model

- Ex: unigram model

$$m(x_{1n}) = \prod_i m(x_i)$$

$$\log m(x_{1n}) = \sum_i \log m(x_i)$$

Perplexity

- Perplexity: $PP(x_{1n}) = 2^{H(L,m)} = m(x_{1n})^{-\frac{1}{N}}$

- Perplexity is the weighted average number of choices a random variable has to make.

- Perplexity is often used to evaluate a language model; lower perplexity is preferred.

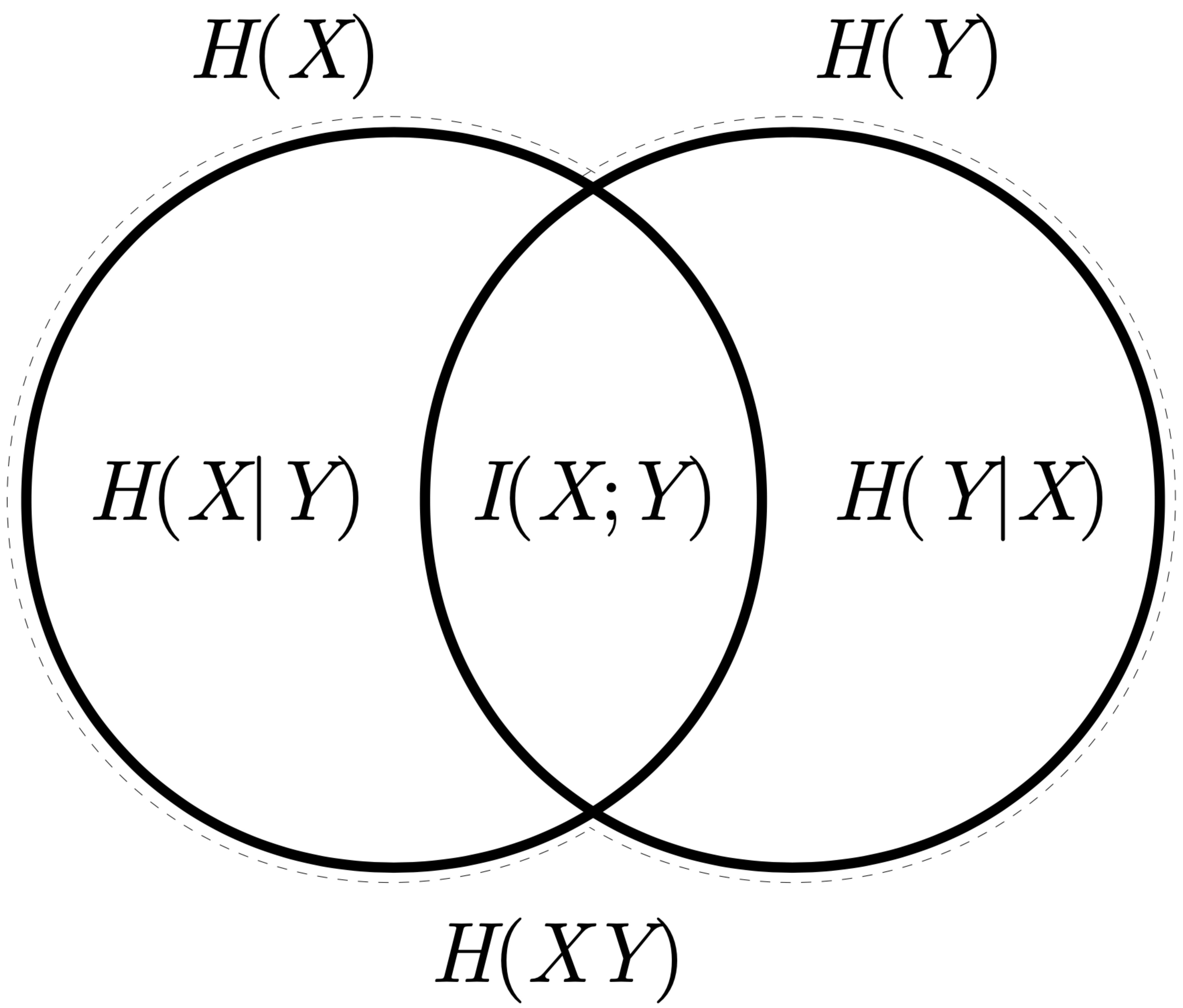
Mutual information

- Measures how much is in common between X and Y:

$$
\begin{aligned}
I(X;Y) &= \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= H(X) + H(Y) - H(X,Y) \\
&= I(Y;X) \\
&= H(X) - H(X|Y) \\
&= H(Y) - H(Y|X)
\end{aligned}
$$

- $I(X;Y) = KL(p(x,y) \| p(x)p(y))$
- If X and Y are independent, I(X;Y) is 0.

The Big Picture



[Dulek and Schaffner 2017](#)
See also Cover+Thomas Fig 2.2; MS Fig 2.6

Summary of Information Theory

- Reading: M&S 2.2 + Cover and Thomas ch 2

- The use of probability theory to quantify and measure “information”.

- Basic concepts:
  - Entropy
  - Cross entropy and relative entropy
  - Joint entropy and conditional entropy
  - Entropy of the language and perplexity
  - Mutual information

Additional slides

Conditional entropy

$$
\begin{aligned}
H(Y \mid X) \\
&= \sum_x p(x) H(Y \mid X = x) \\
&= -\sum_x p(x) \sum_y p(y \mid x) \log p(y \mid x) \\
&= -\sum_x \sum_y p(x, y) \log p(y \mid x) \\
&= -\sum_x \sum_y p(x, y) \log p(x, y) / p(x) \\
&= -\sum_x \sum_y p(x, y)(\log p(x, y) - \log p(x)) \\
&= -\sum_x \sum_y p(x, y) \log p(x, y) + \sum_x \sum_y p(x, y) \log p(x) \\
&= \sum_x \sum_y p(x, y) \log p(x, y) + \sum_x p(x) \log p(x) \\
&= H(X, Y) - H(X)
\end{aligned}
$$

Mutual information

$$
\begin{aligned}
I(X;Y) &= \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= \sum_x \sum_y p(x,y) \log p(x,y) - \sum_x \sum_y p(x,y) \log p(x) - \sum_y \sum_x p(x,y) \log p(y) \\
&= H(X,Y) - \sum_x \log p(x) \sum_y p(x,y) - \sum_y \log p(y) \sum_x p(x,y) \\
&= H(X,Y) - \sum_x (\log p(x)) p(x) - \sum_y (\log p(y)) p(y) \\
&= H(X) + H(Y) - H(X,Y) \\
&= I(Y;X)
\end{aligned}
$$