CS540 Introduction to Artificial Intelligence
Lecture 9

Young Wu
Based on lecture slides by Jerry Zhu and Yingyu Liang

June 20, 2019

Feedback, Lectures

Admin

- Lectures:

1. Concepts: more + RN textbook
2. Examples: more + quiz questions + YouTube videos
3. Applications: more + website and paper links
4. Math: less + YouTube videos
5. Implementation: less + hint file

Feedback, Calculator Question

Admin

- Not allowed $\Rightarrow$ questions will contain expressions without numbers or nice numbers.
- Allowed $\Rightarrow$ questions will contain specific numbers, possibly require rounding etc.
- Yes: $6 + 5 = 11$
- No: $28 + 29 = 57$
- One of: $2 + 0 = 2$

Feedback Calculator

Admin

- Decision: no calculator on both exams.
- You can choose to pay 2 points (out of 40 points) to bring a calculator.
- OR everyone else gets 2 points bonus points for not bringing a calculator.
- Algebra on the exams will be very simple.

Feedback, Formula Sheet

Admin

- A formula sheet will be posted at the beginning of next, next week and distributed with the exam.
- There will be a Piazza post to discuss what formulas you want to add or remove.
- You can choose to pay 2 points for each additional page containing only formulas (including definitions and theorems, i.e. anything on the slides (not annotated)): CANNOT contain any answers to examples, quiz and homework questions.
- Each violation will cost 2 points.

Feedback, Assignments

Admin

- Short instruction: flexibility and creativity
- Long instruction: details, example workflow
- Solutions: details, help with coding
- Each assignment will be re-graded 3 times: all auto-graded, deal with individual submissions after the final no penalty due date.

Discriminative Model vs Generative Model

Review

- Week 1 to Week 4 focus on discriminative models.
- Given a training set $(x_i, y_i)_{i=1}^{n}$, the task is classification (machine learning) or regression (statistics), i.e. finding a function $\hat{f}$ such that given new instances x_i', y can be predicted as $\hat{y}_i = \hat{f}(x_i')$.
- The function $\hat{f}$ is usually represented by parameters w and b. These parameters can be learned by methods such as gradient descent by minimizing some cost objective function.

Perceptron

Review

- Model: LTU Perceptron.
- Objective: minimize mistakes $= \sum_{i=1}^{n} \mathbb{1}_{\{y_i \neq a_i\}}$ or maximize accuracy. It is equivalent to minimizing squared error cost, absolute value cost, log cost (cross entropy loss).
- Training: Perceptron algorithm.
- Prediction: $\hat{y}_i = a_i' = \mathbb{1}_{\left\{w^T x_i' + b \geqslant 0\right\}}$.

Logistic Regression

Review

- Model: Logistic Regression
- Objective: minimize log cost (cross entropy loss) $= \sum_{i=1}^{n} y_i \log(a_i) + (1 - y_i) \log(1 - a_i)$. This is so that the cost is convex in w and b.
- Training: Gradient descent algorithm.
- Prediction:
  $\hat{y}_i = \mathbb{1}_{\{a_i' \geqslant 0.5\}}, a_i' = g\left(w^T x_i' + b\right) = \dfrac{1}{1 + e^{-\left(w^T x_i' + b\right)}}$

Neural Network

Review

- Model: Fully Connected Neural Network
- Objective: minimize squared error cost $= \sum_{i=1}^{n} \left(y_i - a_i^{(L)} \right)^2$.
- Training: Backpropogation: gradient descent algorithm using chain rule.
- Prediction: $\hat{y}_i = \mathbb{1}_{\left\{a'^{(L)} \geqslant 0.5\right\}}, a'^{(l)} = g\left(\left(w^{(l)} \right)^T a'^{(l-1)} + b^{(l)} \right)$ with $a'^{(0)} = x_i'$.

Support Vector Machine

Review

- Model: Support Vector Machine
- Objective: minimize regularized hinge cost $= \sum_{i=1}^{n} \max\left\{0, 1 - (2y_i - 1)\left(w^T x_i + b\right)\right\} + \lambda \|w\|_2^2$ or maximize margin.
- Training: Pegasos algorithm: Primal Estimated sub-GrAdient SOlver for SVM.
- Prediction: $\hat{y}_i = a'_i = \mathbb{1}_{\left\{w^T x'_i + b \geqslant 0\right\}}$.

Decision Tree

Review

- Model: Decision Tree
- Objetive: recursively minimize negative information gain, $H(Y) - H(Y|X_j)$.
- Training: ID3: Iterative Dichotomiser 3.
- Prediction: $\hat{y}_i$ = label of leaf.

Nearest Neighbor

Review

- Model: Nearest Neighbor
- Objective: none.
- Training: memorize the data.
- Prediction: $\hat{y}_i =$ mode $\left\{ y_{(1)}, y_{(2)}, ..., y_{(k)} \right\}$.

Feature Construction

Review

- Each dimension of x_i is a feature, x_{ij}.
- Feature selection is choosing important features to use in predictions: logistic regression regularization, decision tree.
- Feature engineering is creating new features for training: kernelized SVM, convolutional network, traditional computer vision SIFT, HOG, Haar features.

Applications

Review

- All classification tasks.
- Homework 1: Handwritten character recognition.
- Homework 2: Facial expression classification.
- Homework 3: Movie box office prediction.
- Homework 4: Face detection in images.
- All recommendation systems: Amazon, Facebook, Google, Netflix, YouTube ...
- Face recognition, object detection, self-driving cars, speech recognition, spam filtering, fraud detection, weather forecast, sports team selection, algorithmic trading, market analysis, gene sequence classification, medical diagnosis ...

Generative Models

Motivation

- In probability terms, discriminative models are estimating $\mathbb{P}\{Y|X\}$, the conditional distribution. For example, $a_i \approx \mathbb{P}\{y_i = 1|x_i\}$ and $1 - a_i \approx \mathbb{P}\{y_i = 0|x_i\}$.

$a_i > 0.5 \to y = 1$

- Generative models are estimating $\mathbb{P}\{Y, X\}$, the joint distribution.
- Bayes rule is used to perform classification tasks.

digit = 1

$$\mathbb{P}\{Y|X\} = \frac{\mathbb{P}\{Y,X\}}{\mathbb{P}\{X\}} = \frac{\mathbb{P}\{X|Y\}\mathbb{P}\{Y\}}{\mathbb{P}\{X\}}$$

1 2 1

Natural Language

Motivation

- Generative model: ~~next~~ lecture Bayesian network.
- This lecture: a review of probability, application in natural language.
- The goal is to estimate the probabilities of observing a sentence and use it to generate new sentences.

Tokenization

Motivation

- When processing language, documents (called corpus) need to be turned into a sequence of tokens.

1. Split the string by space and punctuations.
2. Remove stopwords such as "the", "of", "a", "with" ...
3. Lower case all characters.
4. Stemming or lemmatization words: make "looks", "looked", "looking" to "look".

Vocabulary

Motivation

- Word token is an occurrence of a word.
- Word type is a unique token as a dictionary entry.
- Vocabulary is the set of word types.
- Characters can be used in place of words as tokens. In this case, the types are "a", "b", ..., "z", " ", and vocabulary is the alphabet.

HW5

Bag of Words Features

Motivation

- Given a document i and vocabulary with size m, let c_{ij} be the count of the word j in the document i for $j = 1, 2, ..., m$.
- Bag of words representation of a document has features that are the count of each word divided by the total number of words in the document.

$$x_{ij} = \frac{c_{ij}}{\sum_{j'=1}^{m} c_{ij'}}$$

fraction of time a word occurs in a doc.

TF IDF Features

Motivation

- Another feature representation is called tf-idf, which stands for normalized term frequency, inverse document frequency.

doc

$$\text{tf}_{ij} = \frac{c_{ij}}{\max_{j'} c_{ij'}}, \quad \text{idf}_j = \log \frac{n}{\sum_{i=1}^{n} \mathbb{1}_{\{c_{ij}>0\}}}$$

# of docs in which j word appeared in

$$\underline{x_{ij} = \text{tf}_{ij}\,\text{idf}_j}$$

- n is the total number of documents and $\sum_{i=1}^{n} \mathbb{1}_{\{c_{ij}>0\}}$ is the number of documents containing word j.

Bag of Words Features Example

Motivation

- Given training set, the set of documents is called a corpus. Suppose the set is "I am Groot", "I am Groot", ... (10 times), "We are Groot". The vocabulary is "I" "am" "Groot" "we" "are", then the bag of words features will have the following training set.

I am Groot

—

—

—

We are Groot

	I	am	Groot	We	are
$x_1 =$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0
$x_2 =$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0
	...	...	...	...	...
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0
x_{11}	0	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Token Notations

Definition

- A word (or character) at position t of a sentence (or string) is denoted as z_t.
- A sentence (or string) with length d is $\underline{(z_1, z_2, ..., z_d)}$.
- $\mathbb{P}\{Z_t = z_t\}$ is the probability of observing $z_t \in \{1, 2, ..., j\}$ at position t of the sentence, usually shortened to $\mathbb{P}\{z_t\}$.

"I" $\longrightarrow$ Pr{"I"} = 0.3

Pr{ Z_0 = "You" $\longrightarrow$ Pr{"You"} = 0.2 }

"Groot" $\longrightarrow$ Pr{"Groot"} = 0

Unigram Model

Definition

- Unigram models assume independence.

$$\mathbb{P}\left\{z_1, z_2, \ldots, z_d\right\} = \prod_{t=1}^{d} \mathbb{P}\left\{z_t\right\}$$

$Pr\{z_1\} \cdot Pr\{z_2\} \cdot \ldots \cdot Pr\{z_d\}$

- In general, two events A and B are indepedent if:

$$\mathbb{P}\left\{A|B\right\} = \mathbb{P}\left\{A\right\} \text{ or } \boxed{\mathbb{P}\left\{A, B\right\} = \mathbb{P}\left\{A\right\}\mathbb{P}\left\{B\right\}}$$

$Pr\{A|B\} = \frac{Pr\{AB\}}{Pr\{B\}} = Pr\{A\}$

- For sequence of words, independence means:

$$\mathbb{P}\left\{z_t|z_{t-1}, z_{t-2}, \ldots, z_1\right\} = \mathbb{P}\left\{z_t\right\}$$

Maximum Likelihood Estimation

Definition

- $\mathbb{P}\left\{z_t\right\}$ can be estimated by the count of the word z_t.

$$\hat{\mathbb{P}}\left\{z_t\right\} = \dfrac{c_{z_t}}{\displaystyle\sum_{z=1}^{m} c_z}$$

Pr { "Groot" }

= # Groot / total # of words.

- This is called the maximum likelihood estimator because it maximizes the probability of observing the sentences in the training set.

Bigram Model

Definition

- Bigram models assume Markov property.

$$\mathbb{P}\left\{z_1, z_2, \ldots, z_d\right\} = \mathbb{P}\left\{z_1\right\} \prod_{t=2}^{d} \mathbb{P}\left\{z_t | z_{t-1}\right\}$$

$= Pr\{z_1\} \cdot Pr\{z_2|z_1\} \cdot Pr\{z_3|z_2\} \ldots Pr\{z_d|z_{d-1}\}$

- Markov property means the distribution of an element in the sequence only depends on the previous element.

$$\mathbb{P}\left\{z_t | z_{t-1}, \boxed{z_{t-2}, \ldots, z_1}\right\} = \mathbb{P}\left\{z_t | z_{t-1}\right\}$$

Conditional Probability

Definition

- In general, the conditional probability of an event A given another event B is the probability of A and B occurring at the same time divided by the probability of event B.

$$\mathbb{P}\left\{A|B\right\} = \frac{\mathbb{P}\left\{AB\right\}}{\mathbb{P}\left\{B\right\}}$$

- For a sequence of words, the conditional probability of observing z_t given z_{t-1} is observed is the probability of observing both divided by the probability of observing z_{t-1} first.

$$\mathbb{P}\left\{z_t|z_{t-1}\right\} = \frac{\mathbb{P}\left\{z_{t-1}, z_t\right\}}{\mathbb{P}\left\{z_{t-1}\right\}}$$

Bigram Model Estimation

Definition

- Using the conditional probability formula, $\mathbb{P}\left\{z_t | z_{t-1}\right\}$, called transition probabilities, can be estimated by counting all bigrams and unigrams.

$$\hat{\mathbb{P}}\left\{z_t | z_{t-1}\right\} = \frac{c_{z_{t-1},z_t}}{c_{z_{t-1}}}$$

$$\frac{\text{count of } \# \; z_{t-1}, z_t}{\text{count of } \# \; z_{t-1}}$$

Unigram MLE Probability

Quiz (Graded)

Q2

- Given the training data "I am Iron Man", "I love you 3000", "I love you mom", "Tell my family I love them", 18 words in total. With the unigram model, what is the probability of observing a new sentence "I love"?
- A: 0
- B: $\frac{3}{18}$
- C: $\frac{3}{4}$
- D: $\frac{4 \cdot 3}{18 \cdot 4}$
- E: $\frac{4 \cdot 3}{18 \cdot 18}$

$$Pr\{\text{"I love"}\} = Pr\{\text{"I"}\} \cdot Pr\{\text{"love"}\}$$

$$\overset{MLE}{=} \frac{\cancel{C_I}\ 4}{18} \cdot \frac{\cancel{C_{love}}\ 3}{18}$$

Bigram MLE Probability, Part I

Quiz (Graded)

- Given the training data "I am Iron Man", "I love you 3000", "I love you mom", "Tell my family I love them", 18 words in total. With the bigram model, what is the probability of observing $Z_2 =$ "love" given the sentence starts with $Z_1 =$ "I"?
- A: 0
- B: $\frac{3}{18}$
- C: $\frac{3}{4}$
- D: $\frac{4 \cdot 3}{18 \cdot 4}$
- E: $\frac{4 \cdot 3}{18 \cdot 18}$

$$\Pr\{\text{"love"} \mid \text{"I"}\} = \frac{C_{\text{"I love"}}}{C_{\text{"I"}}}$$

$$= \frac{3}{4}$$

Bigram MLE Probability, Part II

Quiz (Graded)

Q3

- Given the training data "I am Iron Man", "I love you 3000", "I love you mom", "Tell my family I love them", 18 words in total. With the bigram model, what is the probability of observing a new sentence "I love"?
- A: 0
- B: $\frac{3}{18}$
- C: $\frac{3}{4}$
- D: $\boxed{\frac{4 \cdot 3}{18 \cdot 4}}$
- E: $\frac{4 \cdot 3}{18 \cdot 18}$

$$\Pr\{\text{"I love"}\} = \Pr\{\text{"I"}\} \cdot \Pr\{\text{"love"} \mid \text{"I"}\}$$

$$= \frac{4}{18} \cdot \frac{3}{4}$$

Transition Matrix

Definition

- These probabilities can be stored in a matrix called transition matrix of a Markov Chain. The number on row j column j' is the estimated probability $\hat{\mathbb{P}}\left\{j'|j\right\}$. If there are 3 tokens $\left\{1, 2, 3\right\}$, the transition matrix is the following.

$$\begin{bmatrix} \hat{\mathbb{P}}\left\{1|1\right\} & \hat{\mathbb{P}}\left\{2|1\right\} & \hat{\mathbb{P}}\left\{3|1\right\} \\ \hat{\mathbb{P}}\left\{1|2\right\} & \hat{\mathbb{P}}\left\{2|2\right\} & \hat{\mathbb{P}}\left\{3|2\right\} \\ \hat{\mathbb{P}}\left\{1|3\right\} & \hat{\mathbb{P}}\left\{2|3\right\} & \hat{\mathbb{P}}\left\{3|3\right\} \end{bmatrix}$$

- Given the initial distribution of tokens, the distribution of the next token can be found by multiplying it by the transition probabilities.

Estimating Transition Matrix

Definition

Suppose the vocabulary is "I", "am", "Groot", "we", "are", and the training set contains 10 "I am Groot" then 1 "We are Groot". Then the transition matrix is:

I am Groot I am Groot ... We are Groot

C_I

–	I	am	Groot	we	are
I	$\frac{0+1}{10+5}$ ~~0~~	$\frac{10+1}{10+5}$ 1	0 $\frac{1}{15}$	0 $\frac{1}{15}$	0 $\frac{1}{15}$
am	0	0	1	0	0
Groot	0.9	0	0	0.1	0
we	0	0	0	0	1
are	0	0	1	0	0

$\hat{P}_r\{"am" \mid "I"\}$

Laplace smooth

Trigram Model

Definition

- The same formula can be applied to trigram: sequences of three tokens.

$$\hat{\mathbb{P}}\left\{z_t | z_{t-1}, z_{t-2}\right\} = \frac{c_{z_{t-2},z_{t-1},z_t}}{c_{z_{t-2},z_{t-1}}}$$

- In a document, it is likely that these longer sequences of tokens never appear. In those cases, the probabilities are $\frac{0}{0}$. Because of this, Laplace smoothing adds 1 to all counts.

$$\hat{\mathbb{P}}\left\{z_t | z_{t-1}, z_{t-2}\right\} = \frac{c_{z_{t-2},z_{t-1},z_t} + 1}{c_{z_{t-2},z_{t-1}} + m}$$

add 1 to each count,

size of vocabulary

Laplace Smoothing

Definition

- Laplace smoothing should be used for bigram and unigram models too.

$$\hat{\mathbb{P}}\left\{z_t | z_{t-1}\right\} = \dfrac{c_{z_{t-1}, z_t} + 1}{c_{z_{t-1}} + m}$$

$$\hat{\mathbb{P}}\left\{z_t\right\} = \dfrac{c_{z_t} + 1}{\displaystyle\sum_{z=1}^{m} c_z + m}$$

- Aside: Laplace smoothing can also be used in decision tree training to compute entropy.

Smoothing

Quiz (Graded)

- Fall 2018 Midterm Q12.
- Given a vocabulary of 10^6, a document with 10^{12} tokens with $c_{\text{zoodles}} = 3$. What is the MLE estimation of $\mathbb{P}\{\text{ zoodles }\}$ with and without Laplace smoothing? (choose 2)
- A: $\dfrac{3}{10^{12}}$
- B: $\dfrac{3}{10^6}$
- C: $\dfrac{3+1}{10^{12}+3}$
- D: $\dfrac{3+1}{10^{12}+10^6}$
- E: $\dfrac{3+1}{10^{12}+10^6-1}$

Q5

$P_1\{\text{"zoodles"}\}$

$\dfrac{C_{\text{zoodle}} \ (+1)}{\sum_{\text{words}} C_{\text{words}} \ (+m)}$

$\dfrac{3+1}{10^{12}+10^6}$

$\dfrac{3}{10^{12}}$

N Gram Model

Algorithm

- Input: series $\{z_1, z_2, ..., z_{d_i}\}_{i=1}^{n}$.
- Output: transition probabilities $\hat{\mathbb{P}}\{z_t | z_{t-1}, z_{t-2}, ..., z_{t-N+1}\}$ for all $z_t = 1, 2, ..., m$.
- Compute the transition probabilities using counts and Laplace smoothing.

$$\hat{\mathbb{P}}\{z_t | z_{t-1}, z_{t-2}, ..., z_{t-N+1}\} = \frac{c_{z_{t-N+1}, z_{t-N+2}, \ldots, z_t} + 1}{c_{z_{t-N+1}, z_{t-N+2}, \ldots, z_{t-1}} + m}$$

Sampling from Discrete Distribution

Discussion

- In order to generate new sentences given an N gram model, random realizations need to be generated given the conditional probability distribution.
- Given the first $N-1$ words, $z_1, z_2, ..., z_{N-1}$, the distribution of next word is approximated by $p_x = \hat{\mathbb{P}}\left\{z_N = x \middle| z_{N-1}, z_{N-2}, ..., z_1\right\}$. This process then can be repeated for on $z_2, z_3, ..., z_{N-1}, z_N$ and so on.

Cumulative Distribution Inversion Method, Part I

Discussion

- Most programming languages have a function to generate a random number $u \sim \text{ Unif } [0, 1]$.
- If there are $m = 2$ tokens in total and the conditional probabilities are p and $1 - p$. Then the following distributions are the same.

$$z_N = \begin{cases} 0 & \text{with probability } p \\ 1 & \text{with probability } 1 - p \end{cases} \Leftrightarrow z_N = \begin{cases} 0 & \text{if } 0 \leqslant u \leqslant p \\ 1 & \text{if } p < u \leqslant 1 \end{cases}$$

Cumulative Distribution Inversion Method, Part II

Discussion

- In the general case with m tokens with conditional probabilities $p_1, p_2, ..., p_m$ with $\sum_{j=1}^{m} p_j = 1$. Then the following distributions are the same.

$$z_N = j \text{ with probability } p_j \Leftrightarrow z_N = j \text{ if } \sum_{j'=1}^{j-1} p_{j'} < u \leqslant \sum_{j'=1}^{j} p_{j'}$$

- This can be used to generate a random token from the conditional distribution.

CDF Inversion Method Diagram

Discussion

Sparse Matrix

Discussion

- The transition matrix is too large with mostly zeros.
- Usually, clustering is done so each type (or feature) represent a group of words. For example, instead of having "I", "am", "Groot", "we", "are". Group "I and "we" together and call it type 1, pronoun, group "am" and "are" together and call it type 2, "verb", and leave "Groot" as type 3.
- For the homework, treat each character (letter or space) as a token, then there are $26 + 1$ types. All punctuations are removed or converted to spaces.

27 x 27