CS540 Introduction to Artificial Intelligence
Lecture 9

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

June 16, 2019

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: $\hat{y}_i = \mathbb{1}_{\{a'^{(L)} \geqslant 0.5\}}, a'^{(l)} = g\left(\left(w^{(l)} \right)^T a'^{(l-1)} + b^{(l)} \right)$ with $a'^{(0)} = x_i'$.

Support Vector Machine

Review

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

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 $1 - a_i \approx \mathbb{P}\{y_i = 0|x_i\}$.
- Generative models are estimating $\mathbb{P}\{Y, X\}$, the joint distribution.
- Bayes rule is used to perform classification tasks.

generate image given digit.

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

finding digit given image

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, the words need to be turned into a ~~sequence of features~~ called tokens.

words
characters, letters.

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.

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.

$$\frac{c_{i\text{"Hi"}}}{\text{total \# of words in document.}}$$

$j = \text{"Hi"}$

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

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

TF IDF Features

Motivation

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

$j = 1 \ldots n$

$$\mathrm{tf}_{ij} = \frac{c_{ij}}{\max_{j'} c_{ij'}}$$

"the"

$$\mathrm{idf}_{j} = \log \frac{n}{\sum_{i=1}^{n} \mathbb{1}_{\{c_{ij}>0\}}}$$

→ # documents in which j = "hi" appeared

$$x_{ij} = \mathrm{tf}_{ij}\, \mathrm{idf}_{j}$$

- 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 Characters Features

Quiz (Graded)

Q2

- What is the bag of words feature vector for the string "i am iron man" if the words (tokens) are "i", "a", "m", "r", "o", "n", " "?
- A: $[0, 6, 1, 2, 6, 0, 3, 4, 5, 6, 2, 1, 5]^T$
- B: $\left[\frac{2}{13}, \frac{2}{13}, \frac{2}{13}, \frac{1}{13}, \frac{1}{13}, \frac{2}{13}, \frac{3}{13}\right]^T$
- C: $\left[\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, 1\right]^T$ ← tf
- D: $[2, 2, 2, 1, 1, 2, 3]^T$ ← count
- E: $\left[\frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}\right]^T$

$$\begin{bmatrix} \frac{2}{13} \\ \frac{2}{13} \\ \vdots \end{bmatrix} \quad \frac{c_{ij}}{\sum_{j'=1}^{m} c_{ij'}}$$

Token Notations

Definition

- A word (or character) at position t of a sentence (or string) is denoted as z_t.

z_1 z_2 z_3 z_4

I am iron man

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

token can be any type with diff prob

$Pr\{Z_t = \text{"Hi"}\}$

Unigram Model

Definition

- Unigram models assume independence.

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

$\cdot Pr\{z_3\} \ldots Pr\{z_d\}$

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

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

prob of A given B

dependent

Pr{"love" | "I"}

Pr{"you" | "I"}

- For sequence of words, independence means:

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

Maximum Likelihood Estimation

Definition

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

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

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

MLE Example

Definition

- Let $p = \hat{\mathbb{P}}\{0\}$ in a string with $c_0$0's and $c_1$1's.
- The probability of observing the string is:

$$\binom{c_0}{c_0 + c_1} p^{c_0} (1-p)^{c_1}$$

- The above expression is maximized by:

$$p^\star = \frac{c_0}{c_0 + c_1}$$

MLE Derivation

Definition

characters a, b

a b a b b a b b b. ?

C_a "a" C_b "b"

$Pr\{a\} = p$, $Pr\{b\} = 1-p$

max probability of observing this sample

training set

Unigram → $\arg\max_p \; p(1-p)\,p\,(1-p)(1-p)\,p\,(1-p)(1-p)(1-p)$

$$\arg\max_p \; p^{C_a}(1-p)^{C_b}$$

take log

$$\arg\max_p \; C_a \log p + C_b \log(1-p)$$

set $\frac{\partial L}{\partial p} = 0$

$$\frac{C_a}{p} + \frac{C_b}{1-p} = 0$$

$$C_a - C_a p + C_b p = 0 \Rightarrow \hat{p} = \frac{C_a}{C_a + C_b}$$

Bigram Model

Definition

Pr { "love" | "I" } = 0.2
Pr { "die" | "I" } = 0.1
estimate.

- Bigram models assume Markov property.

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

not independent

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

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

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.

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

- 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.

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

MLE counts

MLE counts

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.

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

~~Uni~~gram MLE Probability

Bi

Quiz (Graded)

- Given the training data "i am iron man", with the unigram model, what is the probability of observing a new string "im"?
- A: 0
- B: $\frac{2}{13}$
- C: $\frac{1}{13 \cdot 13}$
- D: $\frac{2}{13 \cdot 13}$
- E: $\frac{4}{13 \cdot 13}$

$$\Pr\{\text{"im"}\} = \Pr\{\text{"i"}\} \cdot \Pr\{\text{"m"} \mid \text{"i"}\}$$

Bigram.

$$\frac{2}{13}$$

$$\frac{\Pr\{\text{"im"}\}}{\Pr\{\text{"i"}\}} = \frac{0}{\frac{2}{13}}$$

$$= 0$$

Bigram MLE Probability, Part I

Quiz (Graded)

$$\frac{C_{am}}{total}$$

$m \mid a$

- Q3 Given the training data "i am iron man", with the bigram model, what is the probability of observing a new string "~~ia~~" ? am

ignore Q3

- A: 0
- B: $\frac{2}{13}$
- C: $\frac{1}{13 \cdot 13}$
- D: $\frac{2}{13 \cdot 13}$
- E: $\frac{4}{13 \cdot 13}$

$$\Pr\{a\} \cdot \frac{\Pr\{am\}}{\Pr\{a\}} \approx \frac{C_{am}}{C_a}$$

$$\frac{2}{13} \cdot \frac{1}{2} = \frac{1}{13}$$

$$\frac{C_a}{13} \cdot \boxed{\frac{C_{am}}{C_a}}$$

am
an

$m \mid a = \frac{1}{2}$

$n \mid a = \frac{1}{2}$

anything else $\mid a = 0$

ignore 63, 4

Bigram MLE Probability, Part II

Quiz (Graded)

- Given the training data "i am iron man", with the bigram model, what is the probability of observing a new string "am"?
- A: 0
- B: $\dfrac{2}{13}$
- C: $\dfrac{1}{13 \cdot 13}$
- D: $\dfrac{2}{13 \cdot 13}$
- E: $\dfrac{4}{13 \cdot 13}$

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 $\hat{\mathbb{P}}\{j'|j\}$. If there are 3 tokens $\{1, 2, 3\}$, the transition matrix is the following.

27

27

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

bigram transition.

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

Aside: Stationary Probability

Definition

- Given the bigram model, the fraction of times a token occurs for a document with infinite length can be computed. The resulting distribution is called the stationary distribution.

$$p_\infty = p_0^{T} M^\infty$$

$$\begin{pmatrix} \frac{1}{27} \\ \frac{1}{27} \\ \vdots \end{pmatrix}^{T} \begin{pmatrix} \quad \end{pmatrix} = \begin{pmatrix} \quad \end{pmatrix}$$

P_1

Aside: Spectral Decomposition

Definition

- It is easier to find powers of diagonal matrices.
- Let D be the diagonal matrix with eigenvalues of M on the diagonal and P be the matrix with columns being corresponding eigenvectors.

$$MP = \lambda_i P, i = 1, 2, ..., K$$

$$MP = PD$$

$$M = PDP^{-1}$$

$$M^n = \underbrace{PDP^{-1}PDP^{-1}...PDP^{-1}}_{n \text{ times}} = PD^nP^{-1}$$

$$M^\infty = PD^\infty P^{-1}$$

$$\begin{pmatrix} 0.5 & & & \\ & 0.1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

$$||$$

$$\begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

Aside: Stationarity

Definition

- A simpler way to compute the stationary distribution is to solve the equation:

$$p_\infty = p_\infty M$$

Trigram Model

Bigram $\mathbb{P}\{z_t | z_{t-1}\}$

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.

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

a b c

C_{abc}

$\frac{1}{m}$

$C_{\cdot bc}$

Laplace Smoothing

Definition

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

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

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

- 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)

Q6

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

$$\frac{c_{zoodles}+1}{\sum_{j=1}^{m} c_j + m}$$

without Laplace

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

$$\frac{4}{10^{12}+10^6}$$

Bayes Rule

Quiz (Graded)

- Fall 2017 Final Q20
- Two documents A and B. $\hat{\mathbb{P}}\{H\} = 0.1$ in A and $\hat{\mathbb{P}}\{H\} = 0.8$ without Laplace smoothing. One document is taken out at random (with equal probability), and one word is picked out at random (all words with equal probability). The word is H. What is the probability that the document is A?
- A: $\frac{1}{2}$, $B: \frac{1}{3}$, $C: \frac{1}{4}$, $D: \frac{1}{8}$, $E: \frac{1}{9}$

Q8

$\frac{1}{2}$ $\frac{1}{2}$ $Pr\{H|A\}$ $Pr\{H|B\}$

$$Pr\{A|H\} = \frac{Pr\{AH\}}{Pr\{H\}} = \underbrace{\frac{\cancel{Pr\{AH\}}}{Pr\{H|A\} \cdot Pr\{A\} + Pr\{H|B\} \cdot Pr\{B\}}}_{\text{Bayes}}$$

$$= Pr\{H|A\} \cdot Pr\{A\} = 0.1 \cdot \frac{1}{2}$$

$$0.1 \cdot \frac{1}{2} + 0.8 \cdot \frac{1}{2}$$

N Gram Model

Algorithm

- Input: series $\left\{z_1, z_2, ..., z_{d_i}\right\}_{i=1}^{n}$.
- Output: transition probabilities $\hat{\mathbb{P}}\left\{z_t | z_{t-1}, z_{t-2}, ..., z_{t-N+1}\right\}$ for all $z_t = 1, 2, ..., m$.

- Compute the transition probabilities using counts and Laplace smoothing.

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

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 $p_x = \hat{\mathbb{P}}\{z_N = x | z_{N-1}, z_{N-2}, ..., z_1\}$. 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 $K = 2$ 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

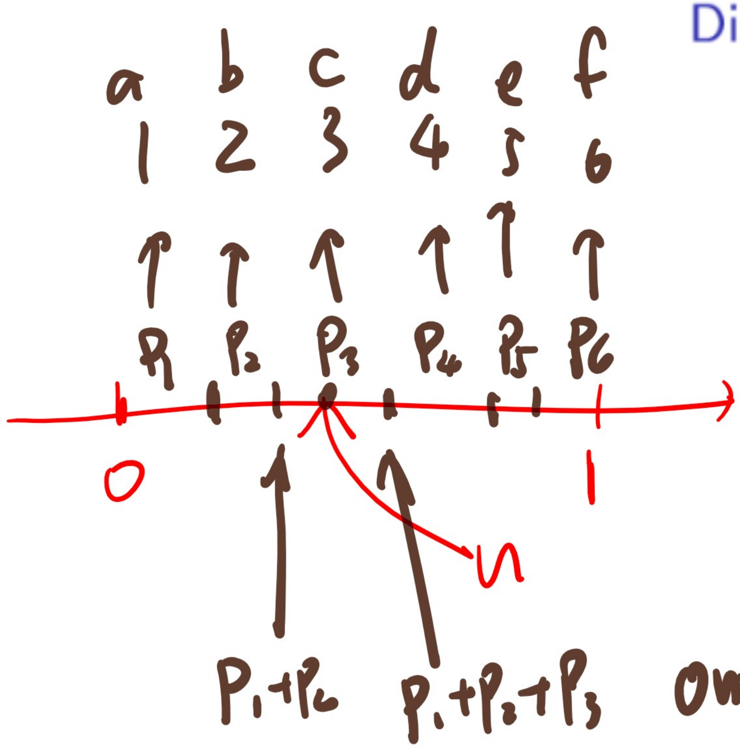
- In the general case with K tokens with conditional probabilities $p_1, p_2, ..., p_K$ with $\sum_{j=1}^{K} p_j = 1$. Then the following distributions are the same.

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

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

CDF Inversion Method Diagram

Discussion



output 3

generate $u \sim \text{Unif}[0, 1]$

if $u < p_1 \longrightarrow$ output 1

$p_1 < u < p_1 + p_2 \longrightarrow$ — 2

$p_1 + p_2 < u < p_1 + p_2 + p_3 \longrightarrow 3$

$\vdots$

Generating New Words

Quiz (Graded)

- Given the transition matrix for characters "i" "a" "m", starting a sentence with the "i" and a uniform random variable $u = 0.5$ is produced. What is the next character?

$$\begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.2 & 0.4 & 0.4 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

- A: "i", B: "a", C: "m"
- D, E: do not choose these.