Lecture 20: Text Model

GU4241/GR5241 Statistical Machine Learning

Linxi Liu

May 3, 2019

Categorical Data

Categorical random variable

We call a random variable ξ **categorical** if it takes values in a finite set, i.e. if $\xi \in (1, \ldots, d)$ for some $d \in \mathbb{N}$. We interpret the d different outcomes as d separate *categories* or classes.

Category probabilities

Suppose we know the probability $t_j = \Pr(\xi = j)$ for each category j. Then

$$t_j \geq 0 \qquad \text{and} \qquad \sum_{j=1}^{d} t_j = 1$$

We can represent the distribution of ξ by the vector $\mathbf{t} = (t_1, \ldots, t_j) \in \mathbb{R}^d$. In other words, we can parameterize distributions of categorical variables by vectors $\mathbf{t}$.

Samples of Size n

A single sample

We can represent a single sample as a vector, e.g.

$$(0,1,0,0,0) \qquad \text{if} \quad d=5 \quad \text{and} \quad \xi=2\ .$$

(Recall the assignments in EM.)

n samples

A sample of size n is a vector of counts, e.g.

$$(2,5,1,3,0)$$

We denote the counts by H_j and write

$$\mathbf{H} := (H_1,\ldots,H_d) \qquad \text{with} \qquad \sum_{j=1}^{d} H_j = n\ .$$

Multinomial Distribution

Modeling assumption

The n observations of ξ are independent, and the probability for $\xi = j$ in each draw is t_j. What is the probability of observing the sample $H = (H_1, \dots, H_j)$?

Multinomial distribution

Answer: The probability is

$$P(\mathbf{H}|\mathbf{t}) = \frac{n!}{H_1! \cdots H_d!} \prod_{j=1}^{d} t_j^{H_j} = \frac{n!}{H_1! \cdots H_d!} \exp\Big(\sum_{j=1}^{d} H_j \log(t_j)\Big)$$

Recall: $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$

Note: The assingment variables M_i in a finite mixture model are multinomially distributed with $n = 1$ and $\theta = (c_1, \dots, c_k)$.

As an exponential family

The form of P above shows that the multinomial is an EFM with

$$S(\mathbf{H}) := \mathbf{H} \qquad h(\mathbf{H}) := \frac{n!}{H_1! \cdots H_d!} \qquad \theta_j := \log t_j \qquad Z(\theta) := 1 \ .$$

Explanation

- In one draw, the probability of observing $\xi = j$ is t_j.
- In n draws, the probability of n times observing $\xi = j$ is t_j^n.

Suppose we have $n = 3$ observation in two categories. How many ways are there to observe exactly two observations in category 1? Three:

	$[1,2]\,[3]$	$[1,3]\,[2]$	$[2,3]\,[1]$
Probability:	$t_1^2 \cdot t_2$	also $t_1^2 \cdot t_2$	again $t_1^2 \cdot t_2$

The total probability of $H_1 = 2$ and $H_2 = 1$ is $3 \cdot t_1^2 \cdot t_2$.

- The number of ways that n elements can be subdivided into d classes with, H_j elements falling into class j, is precisely

$$\frac{n!}{H_1! \cdots H_d!}$$

In the multinomial formula:

$$P(\mathbf{H}|\mathbf{t}) = \underbrace{\frac{n!}{H_1! \cdots H_d!}}_{\#\text{ combinations}} \qquad \underbrace{\prod_{j=1}^{d} t_j^{H_j}}_{\text{probability of } one \text{ combination}}$$

Parameter Estimation

MLE

The maximum likelihood estimator of $\mathbf{t}$ is

$$\hat{\mathbf{t}} = (\hat{t}_1, \ldots, \hat{t}_d) := \frac{1}{n}(H_1, \ldots, H_d) \ .$$

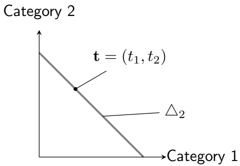
Multinomial Parameters and Simplices

The simplex

The set of possible parameters of a multionmial distribution is

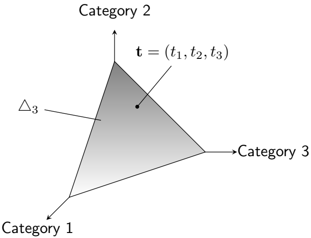
$$\triangle_d := \{\mathbf{t} \in \mathbb{R}^d \,|\, t_j \geq 0 \text{ and } \sum t_j = 1\}$$

$\triangle_d$ is a subset of $\mathbb{R}^d$ and is called the d-**simplex**, or the **standard simplex in** $\mathbb{R}^d$.



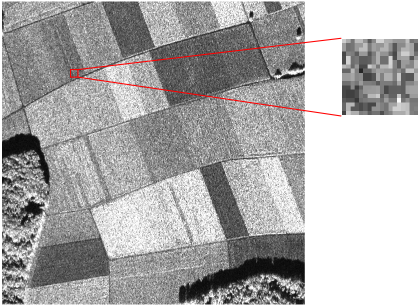
Interpretation

- Each point in e.g. $\triangle_3$ is a distribution on 3 events.
- Each extreme point (corner) correspond to one category j and is the distribution with $t_j = 1$.
- The edges of $\triangle_3$ are the distributions under which only 2 events can occur. (The category corresponding to the opposite corner has zero probability.)
- The inner points are distributions under which all categories can occur.



Example 1: Local Image Histograms

Extracting local image statistics



1. Place a small window (size $l \times l$) around location in image.
2. Extract the pixel values inside the image. If the grayscale values are e.g. $\{0, \dots, 255\}$, we obtain a histogram with 256 categories.
3. Decrease resolution by binning; in Homework 4, we decrease from 256 to 16 categories.

Resulting data

$$\mathbf{H} = (H_1, \dots, H_{16}) \qquad \text{where} \qquad H_j = \# \text{ pixel values in bin } j \ .$$

Since $256/16 = 8$, bin j represents the event

$$\text{pixel value } \in \{(j-1)\cdot 8, \dots, j \cdot 8 - 1\} \ .$$

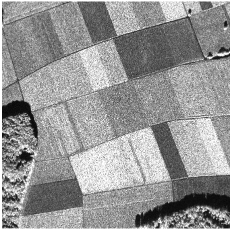
Example 1: Local Image Histograms

Multinomial model

We can model the data by a multinomial distribution $P(\mathbf{H}|\mathbf{t}, n = l^2)$. Then

$$t_j = \text{Pr}\{\xi = j\} = \text{Pr}\{\text{ grayscale value falls in bin } j \text{ }\} .$$

Homework: Multinomial clustering



- The probability of e.g. bin 1 (dark pixels) clearly varies between locations in the image.
- Consequence: A single multinomial distribution is not a good representation of this image.
- In HW 5, the image is represented by a mixture of multinomials which is estimated using EM.

Text Data

Setting

Data set: A huge set of text documents (e.g. all books in a library). The entire set of texts is called a **corpus**.

Can we learn models from text which describe natural language?

Terminology

We have to distinguish occurences of words in a document and *distinct* words in the dictionary. We refer to words regarded as entries of the dictionary as **terms**.

Example 2: Simple Text Model

Data

Suppose our data is a text document. We are given a dictionary which contains all terms occurring in the document.

Documents as vectors of counts

We represent the document as

$$\mathbf{H} = (H_1, \ldots, H_d) \qquad \text{where } H_j = \# \text{ occurences of term } j \text{ in document.}$$

Note:

- d is the number of all terms (distinct words) in the dictionary i.e. d is identical for all documents.
- $n = \sum_j H_j$ can change from document to document.

Example 2: Simple Text Model

Multinomial model

To define a simple probabilistic model of document generation, we can use a multinomial distribution $P(\mathbf{H}|\mathbf{t}, n)$. That means:

- Each word in the document is sampled independently of the other words.
- The probabilities of occurrence are

$$\text{Pr}\{\text{ word } = \text{ term } j\ \} = t_j \ .$$

Implicit assumption

The assumption implicit in this model is that the probability of observing a document is completely determined by how often each term occurs; the order of words does not matter. This is called the **bag-of-words assumption**.

Context

Task

Can we predict the next word in a text?

Context

In language, the co-occurence and order of words is highly informative.
This information is called the **context** of a word.
Example: The English language has over 200,000 words.

- If we choose any word at random, there are over 200,000 possibilities.
- If we want to choose the next word in

  There is an airplane in the __

  the number of possibilities is *much* smaller.

Significance for statistical methods

Context information is well-suited for machine learning: By parsing lots of text, we can record which words occur together and which do not.

The standard models based on this idea are called n*-gram models*.

Bigram Models

Bigram model

A bigram model represents the conditional distribution

$$\text{Pr}(\text{word}|\text{previous word}) =: \text{Pr}(w_l|w_{l-1}) ,$$

where w_l is the lth word in a text.

Representation by multinomial distributions

A bigram model is a *family* of d multinomial distributions, one for each possible previous word.

Estimation

For each term k, find all terms in the corpus which are preceeded by k and record their number of occurences in a vector

$$\mathbf{H}_k = (H_{k1}, \dots, H_{kd}) \text{ where } H_{kj} = \text{number of times term } j \text{ follows on term } k$$

Then compute the maximum likelihood estimate $\hat{\mathbf{t}}_k$ from the sample $\mathbf{H}_k$.
Note: Both j and k run through $\{1, \dots, d\}$.

N-Gram Models

Multinomial representation of bigram

The distributions in the bigram model are:

$$\text{Pr}(\text{word} = j | \text{previous word} = k) = P(H_j = 1 | \hat{\mathbf{t}}_k, n = 1)$$

where P is the multinomial distribution. The entire bigram model is the set

$$\{P(\,.\,| \hat{\mathbf{t}}_k, n = 1) \,|\, k = 1, \dots, d\}$$

N-gram models

More generally, a model conditional on the $(N-1)$ previous words

$$\text{Pr}(w_l | w_{l-1}, \dots, w_{l-(N-1)})$$

is called an N-**gram model** (with the predicted word, there are N words in total).

Unigrams

The special case $N = 1$ (no context information) is the simple multinomial word probability model which we discussed first. This model is also called a **unigram model**.

Learning Shakespeare 1

Unigram Model

To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have

Every enter now severally so, let

Hill he late speaks; or! a more to leg less first you enter

Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

Bigram Model

What means, sir. I confess she? then all sorts, he is trim, captain.

Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.

What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

Enter Menenius, if it so many good direction found'st thou art a strong upon command of fear not a liberal largess given away, Falstaff! Exeunt

From Jurafsky and Martin, "Speech and Language Processing", 2009.

Learning Shakespeare 2

Trigram Model

Sweet prince, Falstaff shall die. Harry of Monmouth's grave.

This shall forbid it should be branded, if renown made it empty.

Indeed the duke; and had a very good friend.

Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

Quadrigram Model

King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;

Will you not tell me who I am?

It cannot be but so.

Indeed the short and the long. Marry, 'tis a noble Lepidus.

From Jurafsky and Martin, "Speech and Language Processing", 2009.

Complexity of N-Gram Models

Enumerating contexts

An N-gram model considers ordered combinations of N terms (=*distinct* words). Say a corpus contains 100,000 words. Then there are

$$100000^N = 10^{5N}$$

possible combinations.

Naive estimate

If we require on average n observations per combination to get a reliable estimate, we would need a corpus containing $n \cdot 10^{5N}$ words.

Consequence

In practice, you typically encountner bigrams or trigrams. Research labs at some internet companies have reported results for higher orders.

Clustering Text

Task

Suppose we have a corpus consisting of two types of text, (1) cheap romantic novels and (2) books on theoretical physics. Can a clustering algorithm with two clusters automatically sort the books according to the two types?

(We will see that there is more to this than solving artificial sorting problems.)

Clustering model

We assume the corpus is generated by a multinomial mixture model of the form

$$\pi(\mathbf{H}) = \sum_{k=1}^{K} c_k P(\mathbf{H}|\mathbf{t}_k) ,$$

i.e. each component $P(\mathbf{H}|\mathbf{t}_k)$ is multionmial.
However: We are now considering **documents** rather than individual words.

Estimation

Apply EM algorithm for multinomial mixture models.

Intepretation: Topics

Thought experiment

Say we run a mixture of two multinomial distributions on the cheap romantic novels and theoretical physics textbooks.

Outcome:

- Each cluster will roughly represent one of the two topics.
- The two parameter vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ represent distributions of words in *texts of the respective topic*.

Word distributions as topics

This motivates the interpretation of clusters as topics.

$$\mathbf{t}_k = \text{ distribution of words that characterizes topic } k$$

Language models derived from this idea are called **topic models**.