Data Processing Basics

Shusen Wang

Processing Categorical Features

Numeric Features and Categorical Features

Age	Gender	Nationality
35	Male	US
31	Male	China
29	Female	India
27	Male	US

Numeric Features and Categorical Features

Age	Gender	Nationality
35	Male	US
31	Male	China
29	Female	India
27	Male	US

- Age is a numeric feature because it is ordered.
- 35-year-old is older than 31-year-old.

Numeric Features and Categorical Features

Age	Gender	Nationality
35	**Male**	US
31	**Male**	China
29	**Female**	India
27	**Male**	US

- Gender is a binary feature: female or male. (In most people’s opinion.)
- Represent ``female’’ by 0.
- Represent ``male’’ by 1.

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	US
31	1	China
29	0	India
27	1	US

- Gender is a binary feature: female or male. (In most people's opinion.)
- Represent ``female'' by 0.
- Represent ``male'' by 1.

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	US
31	1	China
29	0	India
27	1	US

- Nationality is a categorical feature.
- There are 197 countries (arguably.)
- We need to represent countries by numeric vectors.

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	US
31	1	China
29	0	India
27	1	US

Represent countries by numeric vectors.

- First, build a dictionary that maps countries to indices.
- E.g., US→1, China→2, India→3, Japan→4, Germany→5, …
- Count from “1” (instead of “0”).

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	**1**
31	1	**2**
29	0	**3**
27	1	**1**

Represent countries by numeric vectors.

- First, build a dictionary that maps countries to indices.
- E.g., US→1, China→2, India→3, Japan→4, Germany→5, …
- Count from “1” (instead of “0”).

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	**1**
31	1	**2**
29	0	**3**
27	1	**1**

Represent countries by numeric vectors.

- Second, apply one-hot encoding. (Count from “1”.)
- US → 1 → $[1, 0, 0, 0, \cdots, 0]$.
- China → 2 → $[0, 1, 0, 0, \cdots, 0]$.
- $\vdots$

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	$[1, 0, 0, 0, \cdots, 0]$
31	1	$[0, 1, 0, 0, \cdots, 0]$
29	0	$[0, 0, 1, 0, \cdots, 0]$
27	1	$[1, 0, 0, 0, \cdots, 0]$

Represent countries by numeric vectors.

- Second, apply one-hot encoding. (Count from "1".)
- US → 1 → $[1, 0, 0, 0, \cdots, 0]$.
- China → 2 → $[0, 1, 0, 0, \cdots, 0]$.
- $\vdots$

Numeric Features and Categorical Features

Age	Gender	Nationality
35	1	$[1, 0, 0, 0, \cdots, 0]$
31	1	$[0, 1, 0, 0, \cdots, 0]$
29	0	$[0, 0, 1, 0, \cdots, 0]$
27	1	$[1, 0, 0, 0, \cdots, 0]$

Represent countries by numeric vectors.

- Why the indices start from “1” (the US) rather than “0”?
- Reserve “0” (whose one-hot encode is $[0, 0, \cdots, 0]$) for unknown or missing nationalities.

Data Processing

- Represent a person’s feature (age, gender, nationality) using a 199-dim numeric vector.
- For example, convert $(28, \text{Female}, \text{China})$ to vector

$$[28,\ \ 0,\ \ \underbrace{0, 1, 0, 0, \cdots, 0}_{\text{a 197-dim vector for nationality.}}].$$

Data Processing

- Represent a person’s feature (age, gender, nationality) using a 199-dim numeric vector.
- For example, convert $(28, \text{Female}, \text{China})$ to vector

$$[28,\ \ 0,\ \ \underbrace{0, 1, 0, 0, \cdots, 0}_{\text{a 197-dim vector for nationality.}}].$$

- For example, convert $(36, \text{Male}, \text{unknown})$ to vector

$$[36,\ \ 1,\ \ 0, 0, 0, 0, \cdots, 0].$$

Processing Text Data

Step 1: Tokenization (Text to Words)

- We are given a corpus (training data).
  - Corpus is a collection of documents.
  - E.g., all of Shakespeare’s plays.
  - `C[0] = “… to be or not to be that is…”,`
  - `C[1] = “… thus with a kiss i die…”,`
  - ⋮

Step 1: Tokenization (Text to Words)

- We are given a corpus (training data).
  - Corpus is a collection of documents.
  - E.g., all of Shakespeare's plays.
  - `C[0] = "… to be or not to be that is…",`
  - `C[1] = "… thus with a kiss i die…",`
  - ⋮
- Break a piece of text (string) into a list of words, e.g.,
  - `L[0] = [to, be, or, not, to, be, that, is, …],`
  - `L[1] = [thus, with, a, kiss, i, die, …],`
  - ⋮

Step 2: Count Word Frequencies

- Build a dictionary (e.g., hash table) to count words’ frequencies.
- Initially, the dictionary is empty.

Key (word)	Value (frequency)

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

Key (word)	Value (frequency)
`a`	219
`to`	398
`hamlet`	5
`be`	131
`not`	499
`prince`	12
`kill`	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

Key (word)	Value (frequency)
a	219
to	398
hamlet	5
be	131
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

  - Word "to" is in the dictionary.

Key (word)	Value (frequency)
a	219
to	398
hamlet	5
be	131
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

  - Word "to" is in the dictionary.
  - Increase its counter.

Key (word)	Value (frequency)
a	219
to	399
hamlet	5
be	131
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

  - Word “be” is in the dictionary.

Key (word)	Value (frequency)
a	219
to	399
hamlet	5
be	131
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

  - Word "be" is in the dictionary.
  - Increase its counter.

Key (word)	Value (frequency)
a	219
to	399
hamlet	5
be	132
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

  - Word "or" is not in the dictionary.

Key (word)	Value (frequency)
a	219
to	399
hamlet	5
be	132
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Update the dictionary in this way:
  - If word w is **not** in the dictionary, add $(w, 1)$ to the dictionary.
  - If word w is in the dictionary, increase its frequency counter.

- Example:

...	to	be	or	not	to	be	...

  - Word "or" is not in the dictionary.
  - Add ("or", 1) to the dictionary.

Key (word)	Value (frequency)
a	219
to	399
hamlet	5
or	1
be	132
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Sort the table so that the frequency is in the descending order.

Key (word)	Value (frequency)
a	219
to	399
hamlet	5
or	1
be	132
not	499
prince	12
kill	31

Step 2: Count Word Frequencies

- Sort the table so that the frequency is in the descending order.

Key (word)	Value (frequency)
not	499
to	399
a	219
be	132
kill	31
prince	12
hamlet	5
or	1

Step 2: Count Word Frequencies

- Sort the table so that the frequency is in the descending order.
- Replace “frequency” by “index” (starting from 1.)

Key (word)	Value (frequency)
not	499
to	399
a	219
be	131
kill	31
prince	12
hamlet	5
or	1

Step 2: Count Word Frequencies

- Sort the table so that the frequency is in the descending order.
- Replace “frequency” by “index” (starting from 1.)
- The number of unique words is called “vocabulary”.

Key (word)	Value (index)
not	1
to	2
a	3
be	4
kill	5
prince	6
hamlet	7
or	8

Step 3: One-Hot Encoding

- Map every word to its index.
- For example,

```
 Words: [to, be, or, not, to, be]
                ⬇
 Indices: [2,   4,   8,   1,   2,   4]
```

Key (word)	Value (index)
not	1
to	2
a	3
be	4
kill	5
prince	6
hamlet	7
or	8

Step 3: One-Hot Encoding

- Map every word to its index.
- For example,

```
Words: [to, be, or, not, to, be]
                ↓
Indices: [2,  4,  8,  1,  2,  4]
```

- If necessary, convert every index to a one-hot vector.
  - The vectors' dimension is the vocabulary.
  - Vocabulary means # of unique words in the dictionary.

Key (word)	Value (index)
not	1
to	2
a	3
be	4
kill	5
prince	6
hamlet	7
or	8

Step 3: One-Hot Encoding

- If the vocabulary is too big, e.g., greater than 10K, then keep only the 10K most frequent words.
- Why removing infrequent words?

Key (word)	Value (index)
`not`	1
`to`	2
`a`	3
`be`	4
`kill`	5
`prince`	6
`hamlet`	7
`or`	8

Step 3: One-Hot Encoding

- If the vocabulary is too big, e.g., greater than 10K, then keep only the 10K most frequent words.
- Why removing infrequent words?

1. Infrequent words are usually meaningless, e.g.,
   - Name entities, e.g., “Shusen”.
   - Typos, e.g., “prinse” and “hemlat”.
2. Bigger vocabulary → higher-dim one-hot vectors.
   - Slower computation.
   - More parameters in word-embedding layer.

Key (word)	Value (index)
not	1
to	2
a	3
be	4
kill	5
prince	6
hamlet	7
or	8

Step 3: One-Hot Encoding

- If the vocabulary is too big, e.g., greater than 10K, then keep only the 10K most frequent words.
- If a word cannot be found in the dictionary, then simply ignore it.
- Example:

a typo:

```
Words: [to, bi, or]
```

⬇

```
Indices: [2,     8]
```

Key (word)	Value (index)
not	1
to	2
a	3
be	4
kill	5
prince	6
hamlet	7
or	8

Thank you!