Stream Mining

One-Hot Encoding and DGIM

Zeno Adrian Weil

Data Science 1
Goethe University Frankfurt

7th June 2022

One-Hot Encoding

- **categorical** features
  - nominal (e.g. colours)
  - ordinal (e.g. satisfaction levels)
- need for numbers in algorithms
- naïve approach: number serially
  - arbitrary orders
  - meaningless arithmetic calculations
- **one-hot encoding**
  - one binary feature for each possible value

$$x \in \{\text{red}, \text{green}, \text{blue}\} \rightarrow x \in \{1, 2, 3\} \ \text{✗}$$

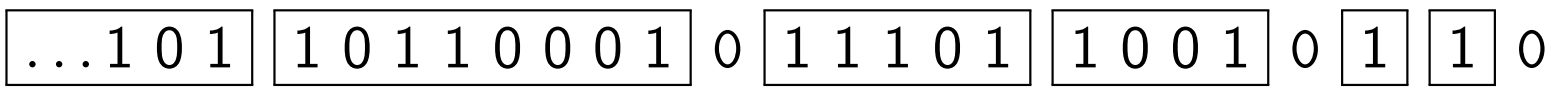
$$x \in \{\text{red}, \text{green}, \text{blue}\} \rightarrow (x_{\text{red}}, x_{\text{green}}, x_{\text{blue}}) \in \{0, 1\}^3 \ \text{✓}$$

The Datar-Gionis-Indyk-Motwani Algorithm

Objectives

- **Estimate** the number of **ones** in a bit stream!
- Be **space-efficient**!

- window size N
- $\mathcal{O}(\log_2 N)$ **buckets**
  - **timestamp**
  - **size** = number of ones
    - powers of two
    - one or two of each size
    - sizes never decreasing moving back
  - include all ones; end with ones
- needs only $\boldsymbol{\mathcal{O}((\log_2 N)^2)}$ **bits**
- **estimation**: half the size of the oldest bucket + sum of sizes of all other buckets
  - error rate: 50%



estimation: 16 reality: 14

The Mushroom Data Set (J.S. Schlimmer, 1987)

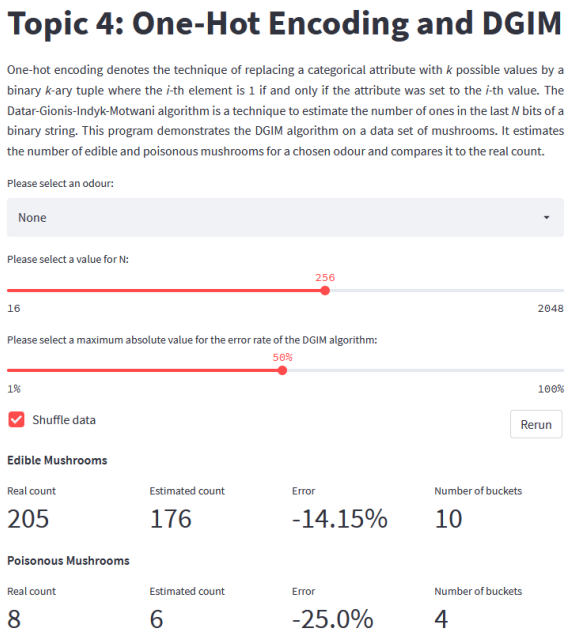
- **8124 samples** of 23 mushroom species
  - 4208 edible
  - 3916 poisonous
- **22 attributes** with 128 possible values
- saved as CSV

```
p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u
e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g
e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m
                        ...
```

Are there simple rules to determine edibility? **Yes!**

The Implementation

- load CSV with Python
- **2D array** for the one-hot encoding of the odour
- Python package **dgim** for the algorithm
- **Streamlit** for the interface
- options
  - odour type
  - window size N
  - error rate



Please select an odour:

None

Please select a value for N:

256

16 2048

Please select a maximum absolute value for the error rate of the DGIM algorithm:

50%

1% 100%

Shuffle data

Rerun

Edible Mushrooms

Real count	Estimated count	Error	Number of buckets
205	176	-14.15%	10

Poisonous Mushrooms

Real count	Estimated count	Error	Number of buckets
8	6	-25.0%	4

Please select an odour:

None

Please select a value for N:

256

16 2048

Please select a maximum absolute value for the error rate of the DGIM algorithm:

50%

1% 100%

Shuffle data

Rerun

Edible Mushrooms

Real count	Estimated count	Error	Number of buckets
214	208	-2.8%	10

Poisonous Mushrooms

Real count	Estimated count	Error	Number of buckets
11	12	9.09%	6

Please select an odour:

None

Please select a value for N:

2048

16 2048

Please select a maximum absolute value for the error rate of the DGIM algorithm:

50%

1% 100%

Shuffle data

Rerun

Edible Mushrooms

Real count	Estimated count	Error	Number of buckets
1675	1872	11.76%	15

Poisonous Mushrooms

Real count	Estimated count	Error	Number of buckets
67	72	7.46%	9

Please select an odour:

None

Please select a value for N:

2048

16 2048

Please select a maximum absolute value for the error rate of the DGIM algorithm:

100%

1% 100%

Shuffle data

Rerun

Edible Mushrooms

Real count	Estimated count	Error	Number of buckets
1651	1872	13.39%	15

Poisonous Mushrooms

Real count	Estimated count	Error	Number of buckets
59	72	22.03%	9

Please select an odour:

None

Please select a value for N:

2048

16 2048

Please select a maximum absolute value for the error rate of the DGIM algorithm:

1%

1% 100%

Shuffle data

Rerun

Edible Mushrooms

Real count	Estimated count	Error	Number of buckets
1678	1672	-0.36%	410

Poisonous Mushrooms

Real count	Estimated count	Error	Number of buckets
68	68	0.0%	68

References

- Project code: `https://github.com/s9770652/DS1-DGIM`
- Mushroom data set: `https://archive-beta.ics.uci.edu/ml/datasets/mushroom`
- Streamlit: `https://streamlit.io/`
- Python package *dgim*: `https://pypi.org/project/dgim/`
- Description of one-hot encoding: `https://sherbold.github.io/intro-to-data-science/04_Data-Analysis-Overview.html#Features`
- Description of the DGIM algorithm (Section 4.6): `http://infolab.stanford.edu/~ullman/mmds/ch4.pdf`