ML Week

0x03 Feature Extraction

Jeff Abrahamson

2–5 novembre 2015

```
>>> from sklearn.feature_extraction import \
    DictVectorizer
>>> onehot_encoder = DictVectorizer()
>>> instances = [
>>>
{’city’: ’New York’},
>>>
{’city’: ’San Francisco’},
>>>
{’city’: ’Chapel Hill’}>>> ]
>>> print onehot_encoder.fit_transform(instances) \
                              .toarray()
[[ 0. 1. 0.] [ 0. 0. 1.][ 1. 0. 0.]]
```

Text features

Bag of words

- Corpus
- Vocabulary
- Words

CountVectorizer

TF - IDF

$$TF_{td} = \frac{f_{td}}{\max_k f_{kd}} \qquad IDF_t = \log_2\left(\frac{N}{n_t}\right)$$

$$TF\text{-}IDF_{td} = TF_{td} \cdot IDF_t$$

with

$$\begin{aligned} f_{td} &= \text{frequency of word (term) } t \text{ in document } d \\ N &= \text{number of documents} \\ n_t &= \text{number of documents containing term } t \end{aligned}$$

SIFT = Scale-Invariant Feature Transformation

D. Lowe, UBC, Distinctive Image Features from Scale-Invariant Keypoints, 2004

SURF = Speeded-Up Robust Features

H. Bay, T. Tuytelaars, and L. Van Gool, SURF: Speeded Up Robust Features, 2006

Questions?

`purple.com/talk-feedback`