CMC MSU Department of Algorithmic Languages
Samsung Moscow Research Center

# Neural Networks for Natural Language Processing

# Нейронные сети в задачах автоматической обработки текстов

*Lecture 2: BOW text representation & Naive Bayes classifier*

Arefyev Nikolay
*CMC MSU Department of Algorithmic Languages &
Samsung Moscow Research Center*

# Naive Bayes classifier

- **Classification** / regression / clustering / …
  - Outputs are from finite set (called classes or labels)

    unlike regression, where ???
  - Classes are known in advance

    unlike clustering, where ???
- **Supervised** / unsupervised / reinforcement / …
  - Trains on a train set: a set of $\mathbf{(x_i,y_i)}$ pairs

    the difference with unsupervised / reinforcement learning?
  - The train set contains examples for all possible classes.
- **Probabilistic** / non-probabilistic
  - Estimates the probability distribution $P(y|x)$: the probabilities that a given example belongs to each possible class $y$ in $\{c_1,...c_K\}$
  - They will sum to 1.
  - For binary classification (2 classes) we usually estimate only $P(y=1|x)$. And $P(y=0|x) = 1-P(y=1|x)$

# When do we need a classifier?

- Text classification
  - Spam detection (binary: spam, not spam)
  - Topic categorization (K classes: sport, art, politics, ...)
  - Sentiment analysis (positive, negative, ?neutral?)

- Text tagging:
  classify each token in a given text
  - Part of speech (POS) tagging
  - Named Entity Recognition (NER)

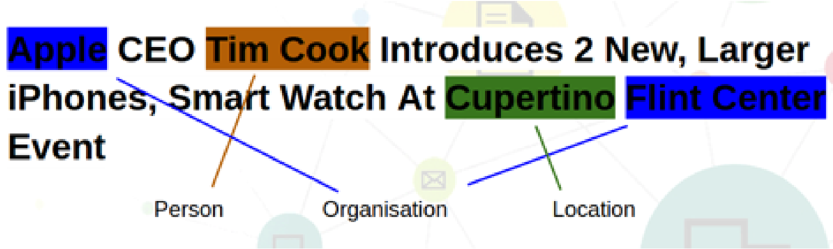


*Figure 1:* An example of NER application on an example text
From https://ru.bmstu.wiki/NER_(Named-Entity_Recognition)

- [Conditional] Text generation:
  generate word by word, sampling from predicted distribution over possible next tokens $P(w_i|w_{i-1},w_{i-2},\ldots,w_1,[COND])$
  - Machine Translation (COND – source text)
  - Chat bots (COND – dialog history)
  - Image captioning (COND – picture)

# Relative frequencies

Given a train set of documents and their classes $D_{train}=\{(d_i,c_i)\}$
what is <u>the simplest way</u> to estimate ***P(c|d)****?*

*P(c=pos|d='Total trash.') = ?*

But we want to work on <u>new documents</u>, not those from the train set:

*P(c=pos|d='This movie was not very good. Though a few funny moments made me laugh, I will not recommend it to anybody. ') = ?*

# Bayes classifier

$$P(c|d)=\frac{P(d|c)P(c)}{P(d)}$$

$$\underset{k}{\operatorname{argmax}}\, P(c_k|d)$$

# Generative model

We need a (probabilistic) model of document generation: what kind of documents can be generate for each given class?

$P(d|c)$

To simplify, we will treat a document as **a bag of words (BOW)**

- Bag (multiset) vs. set vs. list
  - 'I love cats. When I was born, I saw two cats.' {I:3, love:1, cats:2, when:1, ...}
- trivially extends to bag of ngrams (BON)
  - 'Food was not bad. I will return'
  - 'Food was bad. I will not return'

# Bernoulli Naive Bayes


Document is multivariate random variable $V=(v_1,\ldots,v_M)$

- $v_i \sim Bernoulli(p_{ik})$ ← independent, but not identically distributed
- A document can be represented as M-dimensional vector of 0 and 1
- Word counts are ignored.
  ‘I love cats very much’ = ‘I love cats very very much’

P(d|c) = … ← DERIVE

# Multinomial Naive Bayes

Icosaèdre inscrit, en bronze.
*(Collection de S. M. le roi Fouad Ier).*

Class k

$w_1$ $w_2$ $w_3$ $w_4$ ............ $w_N$

Document is generated by

- Sampling length N~P(n) from some distribution
- Throwing k-th dice N times

- A document can be represented as M-dimensional BOW or binary BOW vector.
  - Word order is ignored.
  - Word counts can be ignored or not.

P(d|c) = … ← DERIVE

# Alpha smoothing

What if some word was not in the training examples of $c_j$?

$$\widetilde{P}(w_i|c_j)=0 \Rightarrow \widetilde{P}(c_j)\prod_{i=1\ldots N}\widetilde{P}(w_i|c_j)=0$$

Problem!

Hack: let's add "pseudo-counts"

$$\widetilde{P}(w_i|c_j)=\frac{\alpha+\sum_{d\in c_j} occurences\,of\,w_i}{\alpha M+\sum_{d\in c_j} all\,words}$$

# Log trick

- Multiplication of many small numbers leads to underflow!

$$\widetilde{P}(c_i) \prod_{j=1\ldots N} \widetilde{P}(w_j|c_i)$$

- Sum log probabilities instead.

$$\log \prod_{j=1\ldots N} \widetilde{P}(w_j|c_i) = \sum \log \widetilde{P}(w_j|c_i)$$