Lesson 2 : Word embeddings

(or how to make sense of

King - Man + Woman ≈ Queen

when you are a machine)

In this lesson, we make a huge leap forward in time from 1913 (Markov) to 2013 (Mikolov et al)

- Reminder et al. ≡ "et alia" meaning "and others".
- Flash article on screen.

This article is the basis of a miraculous algorithm called word2vec (code.google.com / archive/p / word2vec/)

My choice of wording is careful :

"Miraculous" ≡ Incredible results but we do not understand why.

Indeed, as we shall see, the ~~definition~~ justification of the underlying statistical model is not at the mathematical level of rigor yet the technique works incredibly well.

I think it is important to ~~have~~ know that, while keeping an open mind

General idea: Starting from a (large) corpus of text, Word2Vec constructs an embedding $\hat{\varphi}$: Words $\longrightarrow \mathbb{R}^d$ $\quad d >> 1$

$$w \longmapsto \varphi(w) = v_w$$

Here $\mathbb{R}^d$ is "the vector space of features" and $\hat{\varphi}$ is constructed by training a model which tries to predict a word w according to its context i.e neighboring words.

The space parameter Θ is the set of maps φ

$$\hat{\varphi} = \underset{\varphi \in \Theta}{\text{Argmax}} \; \log L(\varphi, \overset{\text{Text}}{\text{Corpus}})$$

$\log L \hookrightarrow$ log Likelihood

More precisely: Although there are variants and multiples additionnal tricks, let us detail the standard CBOW Model (Continuous Bag-of-Words)

Given a corpus made of an ordered set of words $(w_1, w_2, \dots, w_T)$; there are multiple assumptio

1/ We assume that the log-likelihood is of the form $L = \sum_{t=1}^{T} \log(w_t \mid C_t)$ $\quad C_t \hookrightarrow$ "Context of t-th word"

$$C_t = \{ \omega_{t+k} \mid 1 \leq |k| \leq c \}$$

$$|C_t| = 2c$$

Remark: As we read the text, this can be seen as if the appearance of words is Markovian with the context as the single explanatory variable

2/ Assume there is a scoring function $S_\varphi(\omega, C) = s(\omega, C)$ which indicates the score of a word ω appearing in context C. ⚠ This is where enters φ

We take as definition:

$$s(\omega, C) = \frac{1}{|C|} \sum_{\omega' \in C} \langle \underbrace{\varphi(\omega)}_{v_\omega}, \underbrace{\varphi(\omega')}_{v_{\omega'}} \rangle = S_\varphi(\omega, C)$$

3/ We take for $\log p(\omega | C)$ the following (↗ with score of positive outcomes ; ↘ with score of negative outcomes)

$$\log p(\omega | C) = \log\left(1 + e^{-s(\omega, C)}\right) + \sum_{n \in N_C} \log\left(1 + e^{-s(\omega, C)}\right)$$

N_C = set of sampled negative ~~samples~~ candidates

i.e words which never appear with C

Indeed, as much it is important to teach the machine what are the positive outcomes, it is also important to teach the negative ones!

In the end,

$$\log L(\varphi) = \sum_{t=1}^{T} \left[\log\left(1 + e^{S_\varphi(w_t, C_t)}\right) + \sum_{ne \, N_{C_t}} \log\left(1 + e^{-S_\varphi(w_t, C_t)}\right) \right]$$

$$\rightsquigarrow \quad \hat{\varphi} = \underset{\varphi}{\text{Argmax}} \; \log L(\varphi)$$

↳ Learned through gradient descent.