DEMS publications topic modelling

Alessandro Asperti 813224, Sofia Davoli 813479

DEMS publications dataset

Dataset list_20200522 contains information about publication of DEMS Bicocca's departement till 2020. It contains 272 columns. Columns about title, authors, abstract and key words are of our interest. We will use this information to perform a topic modeling analysis.

```
## [1] 2922  272
```

PREPROCESSING

Let us build a data frame (tibble object) merging publication title, keywords, abstract, journal name in one string. We want to keep also the publication ID and Scopus' subject classification.

We also want to select records that contain only *journal articles* written in *English* and exclude those publications that do not have abstracts.

text	autori	id
Length:868	Length:868	Min. : 235
Class :character	Class :character	1st Qu.: 43237
Mode :character	Mode :character	Median :175173
NA	NA	Mean :155773
NA	NA	3rd Qu.:257612
NA	NA	Max. :331324

Let us make a new data frame where:

1. each word gets in a different row;
2. stop words are deleted (stop words are very common words such as articles, function words, . . .);
3. characters that are not literals are eliminated;
4. words stemming (finding the root of each word) using wordStem function

	autori	id	word
Len	gth:85261 Min	. : 235 Len	gth:85261
Cla	ss :character 1st	Qu.: 61530 Cla	ss :character
Mod	e :character Med	ian :183590 Mod	e :character
NA	Mea	n :165778 NA	
NA	3rd	Qu.:264792 NA	
NA	Max	. :331324 NA	

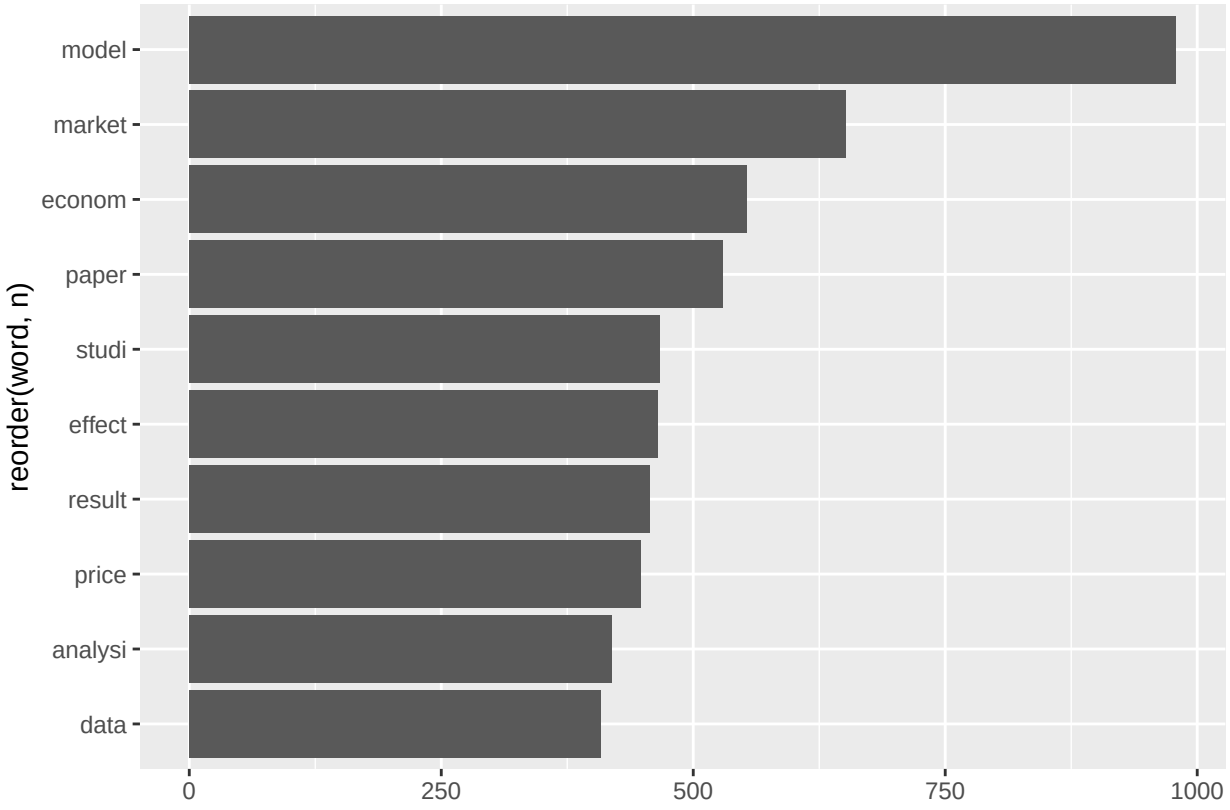
Notice that:

1. `unnest_tokens` takes the string in the `input` filed and created a dataset with a row for each word in the field passed to the `output` argument;
2. the `stop_words` tibble is contined in the *SnowballC* package and that `anti_join()` keeps the rows of the first dataset that are not in the second dataset (according to the field(s) indicated in the `by` argument);
3. `str_extract()` extracts the substring compilant with the regular expression passed as second argument; the meaning of `[a-z']+` only letters from "a" to "z", the apostroph "'" that appear one or more ("+") times;
4. the function `wordStem()` that carries out the stemming belongs to the `SnowballC` package.

Let us count the words and order decreasingly.

word	n	perc
na	1660	1.9469629
NA	1639	1.9223326
model	979	1.1482389
market	651	0.7635378
econom	553	0.6485967
paper	529	0.6204478
studi	467	0.5477299
effect	465	0.5453842
result	457	0.5360012
price	448	0.5254454

We can notice that that there are many "na" strings and many missing values. It looks like at some points missing values have been cast into "na" strings. Let us eliminate both and recompute the tibbles.



We want to build a data frame with count per word per publication.

id	word	n
261901	energi	24
3621	oil	23
265552	skill	21
261901	intens	18
3637	condit	17
142394	fire	17
74650	train	16
132948	punish	16
228978	region	16
235003	fdi	16

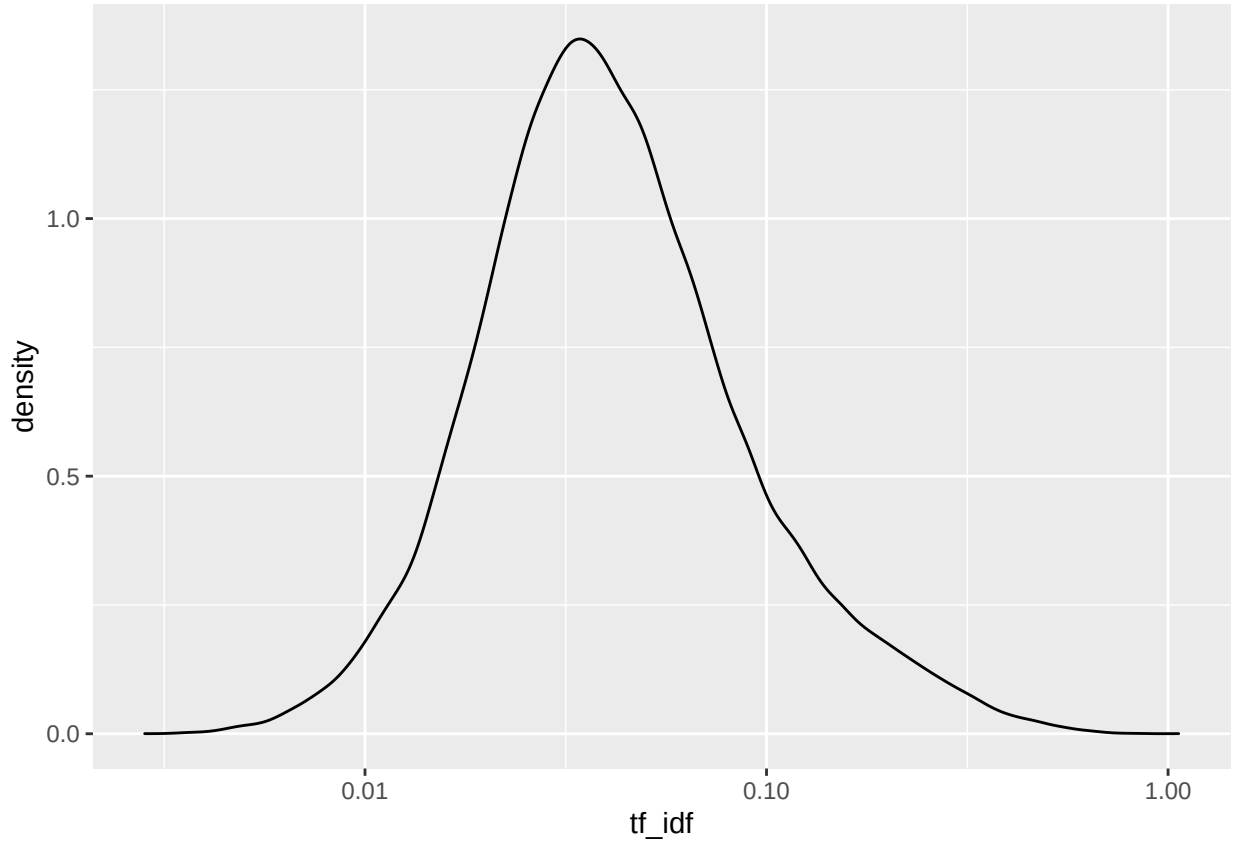
Since some words are very frequent in few documents (very discriminant for clustering) while others are present in many documents (not discriminant for clustering), a measure that try to compensate these extremes is the *tf-idf*, where *tf* stands for term frequency, while *idf* stands for inverse document frequancy and is computed as

$$idf = \log\left(\frac{\text{n. of documents}}{\text{n. of docs in which term is present}}\right)$$

The *tf_idf* quantity is obtained as product of the document frequency times the *idf*. Let us use the *tidytext* function to add the *tf-idf* computations to the dataset.

Let us check how this quantity is distributed.

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.002826 0.024864 0.038714 0.056838 0.064417 1.062783
```



Normally, for document clustering and topic modelling is carried out after exluding words with low *tf-idf*. The first quartile or even the median are possible cut-offs. Here median cut-off is applied.

Now, let us build a document-term matrix to compute similarity measures between documents or apply the Latent Dirichlet Allocation method.

We can use the `spread()` function in the *tidyr* package:

word	235	340	423	538
a	0	0	0	0
aaa	0	0	0	0
aasri	0	0	0	0
abandon	0	0	0	0
abat	0	0	0	0

This matrix is typically very sparse (with many zeros):

```
## [1] 0.006176676
```

In fact, in this case we have less than 1% of non-zero values.

A possible similarity matrix can be based on the product of the number of equivalent words in different documents. This measure can be cast into 0-1 as in the convertion of a covariance matrix into a correlation matrix. A distance can be built as $1 - \rho$, where ρ is the correlation-like measure.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.07715 0.98551 1.00000 0.98050 1.00000 1.00000
```

Most documents tend to be very distant from each other (at distance 1): they probably have no words in common. Maybe abstracts are a too short to be used in this context. Thus, we cannot expect a successful application of hierarchical clustering.

However, the Latent Dirichlet Allocation method can be more succesfull. The *topicmodels* package's `LDA()` function needs a document-term matrix in the format returned by the `tm` package. We can use the `cast_dtm()` function to build it.

```
## <<DocumentTermMatrix (documents: 868, terms: 5858)>>
## Non-/sparse entries: 25585/5059159
## Sparsity           : 99%
## Maximal term length: 34
## Weighting          : term frequency (tf)
```

The *textmineR* package's `FitLdaModel` function needs a document-term matrix of class dgCMatrix. We can use the `CreateDtm` function to build it.

CHOOSING NUMBER OF TOPIC

Once the document-term matrix is been created and the LDA applied, we need to choose k number of topics.

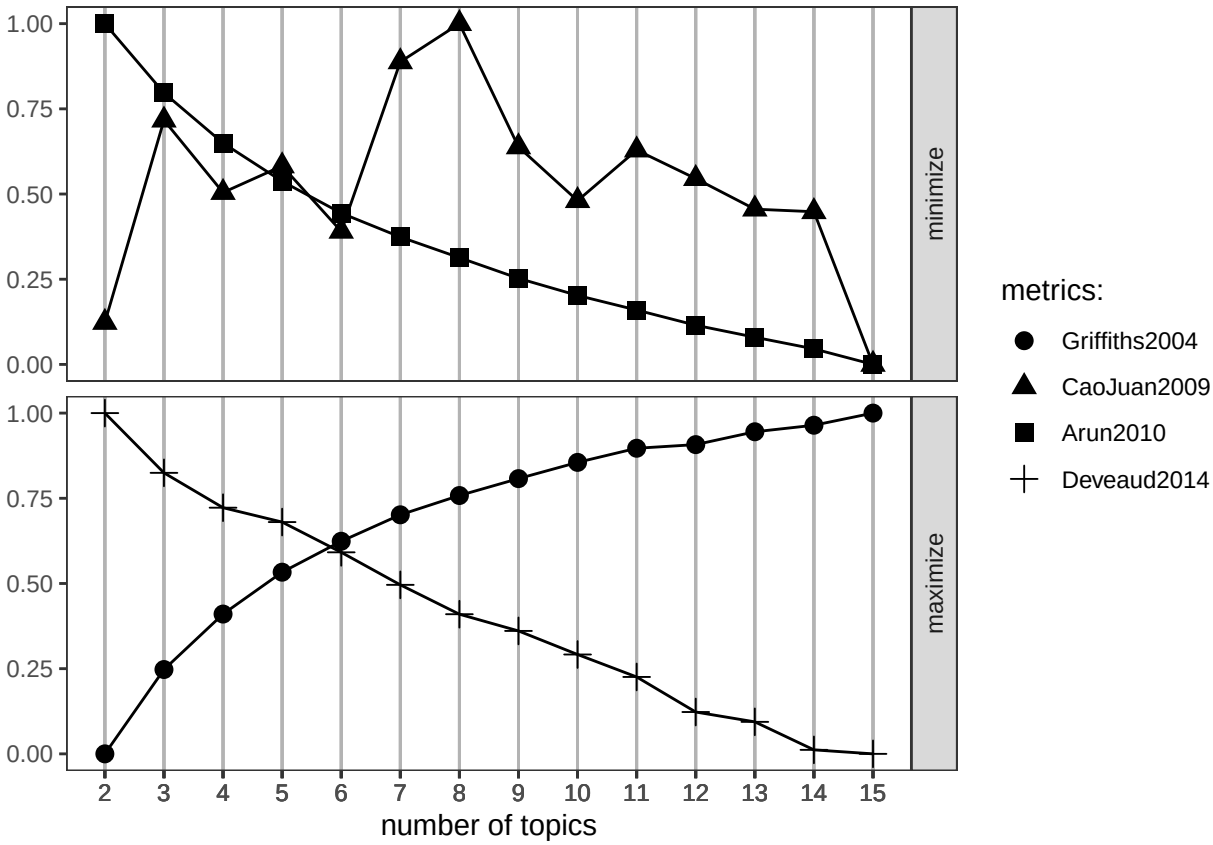
Although there is no single, uniform measure for choosing the number of topics in building a topic model, several methods have been proposed to help decide on the number of topics k.

Two methods (Coa Juan and Arun) aim to minimize the metrics to determine the optimal number of topics. Both Coa Juan and Arun use measures of distance to make decisions regarding k. The other two methods (Deveau and Griffiths) aim to maximize the metrics to determine the optimal number of topics.

- Cao Juan 2009 uses minimum density measures to choose the number of topics.
- Arun 2010 utilize a measure of divergence, where minimal divergence within a topic is preferred.
- Deveaud 2014 utilize a measure maximizing the divergence across topics.
- Griffiths2004 maximize the log-likelihood of the data over different value of k.

We use these four measures across 2-15 topics.

```
## fit models... done.
## calculate metrics:
##   Griffiths2004... done.
##   CaoJuan2009... done.
##   Arun2010... done.
##   Deveaud2014... done.
```



The Plot show that the optimal number of topic is 6 (in both cases the 2 lines intersect at 6).

Once the optimal number of topic is found, there's the need of human intuition to decide wether the optimal number of topics is sufficent to create topics.

Assessing number of topics

We used 2 different LDA model estimator: - LDA function which implements VEM algorithm. This typeof object allows us to compute the per-document-per-topic probability.

- FitLdaModel function which implements Gibbs sampling algorithm. This function contains a parameter that allows us to compute topic coherence. We use both model to assess number of topic.

```
## top6
##   1   2   3   4   5   6
## 188 138 123 140 171 108
```

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
market	model	dynam	social	bank	analysi
price	estim	agent	effect	polici	function
competit	condit	heterogen	incom	financi	water
oil	test	bifurc	firm	care	network
product	statist	stabil	distribut	firm	optim
shock	price	ration	process	cost	demand
global	sampl	patient	inequ	monetari	product

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
electr	rate	equilibrium	random	corpor	relat
inequ	correl	mechan	chang	public	patent
manag	volatil	job	measur	govern	set

```
##        t_1        t_2        t_3        t_4        t_5        t_6
## 0.14509207 0.07007903 0.08898191 0.17021532 0.15408902 0.27791796
```

t_1	t_2	t_3	t_4	t_5	t_6
paper	management	data	model	price	patients
inequality	global	model	dynamics	financial	health
countries	market	based	agents	oil	care
income	analysis	approach	equilibrium	market	data
economic	innovation	models	stability	policy	study
data	paper	random	competition	model	clinical
level	social	analysis	show	prices	risk
evidence	research	statistics	heterogeneous	models	results
firms	network	distribution	preferences	shocks	analysis
effect	corporate	paper	market	markets	brain

From both model's top 10 word per topic it is difficult to define the topic. We think that 6 topic are not enough to select the definition of the same.

So we try to apply the LDA with 7 topics.

```
## top7
##   1   2   3   4   5   6   7
## 122  61 155 141 118 165 106
```

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
estim	patient	polici	dynam	analysi	firm	price
test	care	social	bank	skill	countri	model
sampl	health	market	agent	inform	effect	oil
random	integr	global	heterogen	job	product	market
function	clinic	manag	financi	educ	invest	shock
process	cost	rule	stabil	prefer	italian	volatil
distribut	age	competit	bifurc	satisfact	region	energi
bayesian	control	design	market	index	perform	electr
model	contract	network	game	wage	intern	condit
set	diseas	optim	ration	labour	tax	correl

We obtain the following topic: topic 1 = statistic topic 2 = health topic 3 = political economy topic 4 = mathematical economics topic 5 = labour economics topic 6 = industrial economics topic 7 = energy

t_1	t_2	t_3	t_4	t_5	t_6	t_7
data	analysis	paper	model	management	price	patients
model	risk	inequality	dynamics	market	oil	care
based	data	economic	agents	global	market	health

t_1	t_2	t_3	t_4	t_5	t_6	t_7
models	environmental	countries	equilibrium	innovation	financial	clinical
approach	model	evidence	stability	social	prices	study
random	multi	income	competition	paper	policy	results
analysis	design	firms	show	knowledge	shocks	data
statistics	italy	data	game	corporate	model	brain
distribution	time	level	rule	information	markets	state
paper	models	economics	market	research	models	subjects

Both models used to fit LDA with 7 topic gave as resulting top 10 word which distingue better the 7 topic. Topics defined using topicmodel package reappear in textmieR result, in fact top 10 words for topic are almost the same for the 2 models. we can compute coherence of topic, thaks to textmineR package and assume that this results can be considered as significant also for the other model.

```
##        t_1        t_2        t_3        t_4        t_5        t_6        t_7
## 0.08841313 0.06144966 0.09844367 0.17021532 0.04585756 0.24194945 0.39703792
```

Obtained result suggest that the 7 topic are not really coherent. We try to evaluate the topics using human judgement technique (we compute the per-document-per-topic probabilities and than we analyze the texts) .

GAMMA PROBABILITY

We can examine the per-document-per-topic probabilities, called gamma, with the matrix = "gamma" argument to tidy().

document	topic	gamma
298431	1	0.0008467
42614	1	0.1363310
12556	1	0.0012340
228980	1	0.0005756
6836	1	0.0005756
31381	1	0.0008934
11554	1	0.0005860
265560	1	0.0005201
139303	1	0.9931204
298413	1	0.0008467
228975	1	0.0005039
331324	1	0.5150788
15304	1	0.0009456
244060	1	0.5562599
44732	1	0.0015247
261906	1	0.0004962
170329	1	0.0004675
12348	1	0.0011074
279624	1	0.0716216
309800	1	0.2611043

Each of these values (gamma) is an estimated proportion of words from that document that are generated

from that topic. For example, the model estimates that about 99,31% of the words in document 139303 were generated from topic 1. To confirm this result, we checked what the most common words in document 139303 were:

```
## # A tibble: 17 x 3
##    document term         count
##    <chr>    <chr>        <dbl>
##  1 139303   beta             4
##  2 139303   central          4
##  3 139303   distribut        4
##  4 139303   represent        2
##  5 139303   moment           2
##  6 139303   mixtur           1
##  7 139303   function         1
##  8 139303   hypergeometr     1
##  9 139303   defin            1
## 10 139303   tractabl         1
## 11 139303   express          1
## 12 139303   properti         1
## 13 139303   statistica       1
## 14 139303   interpret        1
## 15 139303   standard         1
## 16 139303   deriv            1
## 17 139303   includ           1
```

This appears to be an article about statistic. Which means that the algorithm was right to place this document in topic 1.

KOHONEN MAP

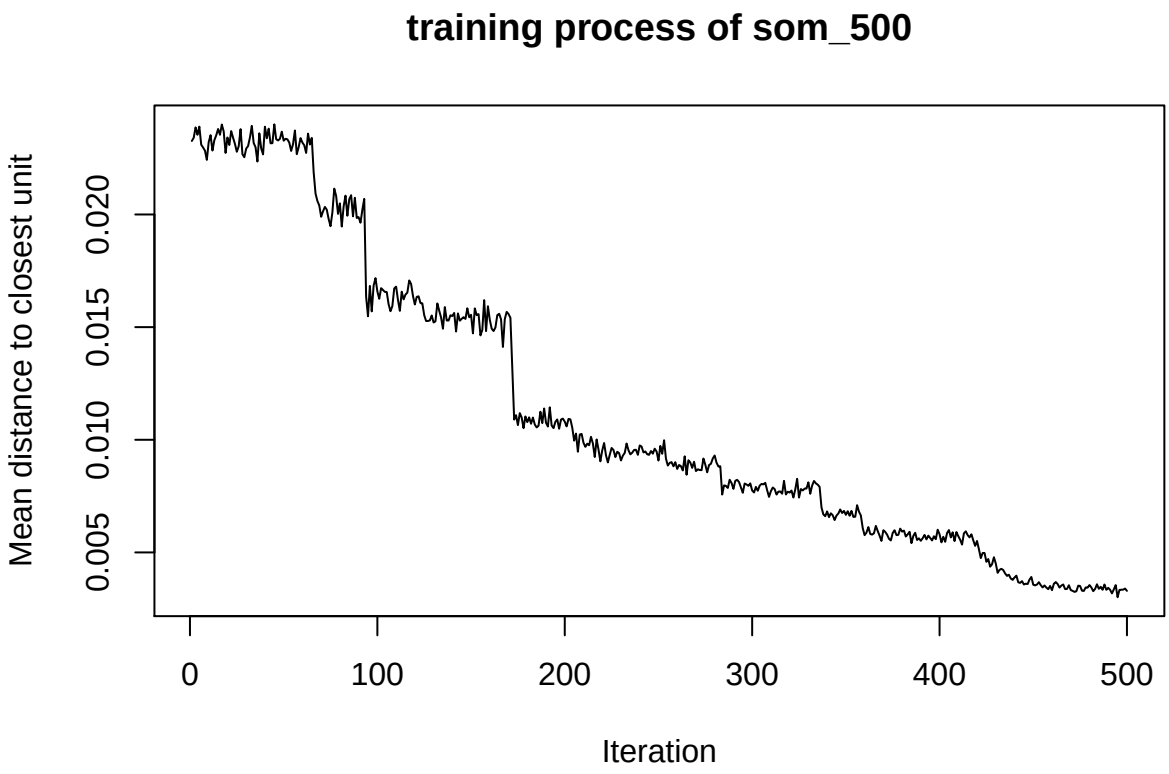
Kohonen map or SOM(Self Organizing Map) is a method to do dimensionality reduction. They use a neighborhood function to preserve the topological properties of the input space. This makes SOMs useful for visualization and for understanding patterns and caractherists like correlations between topics and data.

```
## # A tibble: 6 x 9
##   document  topic_1  topic_2  topic_3  topic_4  topic_5  topic_6  topic_7   Sum
##   <chr>       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl>
## 1 100600   0.000468 0.000468 0.000468 0.765    0.232    0.000468 0.000468  1
## 2 101172   0.000586 0.000586 0.996    0.000586 0.000586 0.000586 0.000586  1
## 3 101467   0.000468 0.000468 0.000468 0.000468 0.000468 0.000468 0.997     1
## 4 101483   0.000448 0.386    0.000448 0.000448 0.000448 0.000448 0.612     1.00
## 5 102048   0.000671 0.000671 0.000671 0.000671 0.357    0.640    0.000671  1
## 6 103208   0.000586 0.000586 0.000586 0.000586 0.394    0.603    0.000586  1
```

We created 2 SOM model, one with 500 iterations and one with 1000 iterations to check which one convergence.

Changing Line: Progression of the learning process.

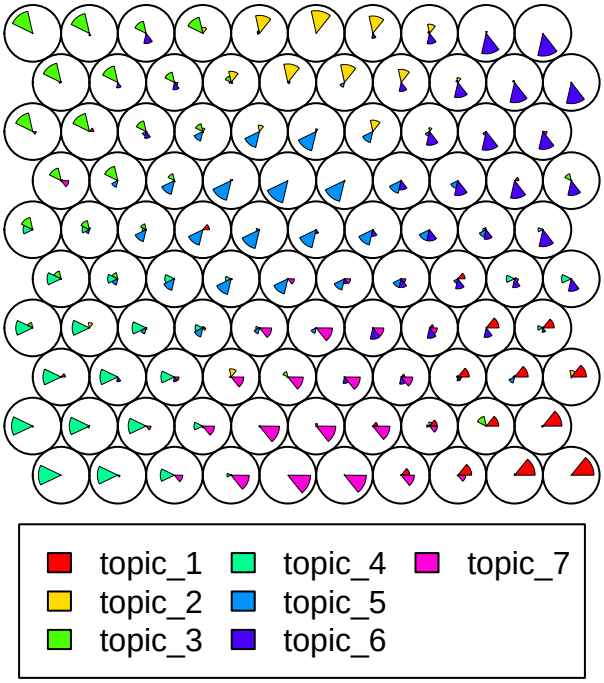
This graph enables to appreciate the convergence of the algorithm. It shows the evolution of the average distance to the nearest cells in the map. If there appear a fast decreasing, the number of iteration can be minimized. By default, the procedure requests RLEN = 100 iterations.


training process of som_500
Mean distance to closest unit
0.005
0.010
0.015
0.020
0
100
200
300
400
500
Iteration

For our dataset we have a slow decreasing but it converge to zero. It means that we need at least 500 iterations. We check if with 1000 iterations we can have a faster convergence, but this doesn't happen.

Codes Book: distribution of argument in the plot

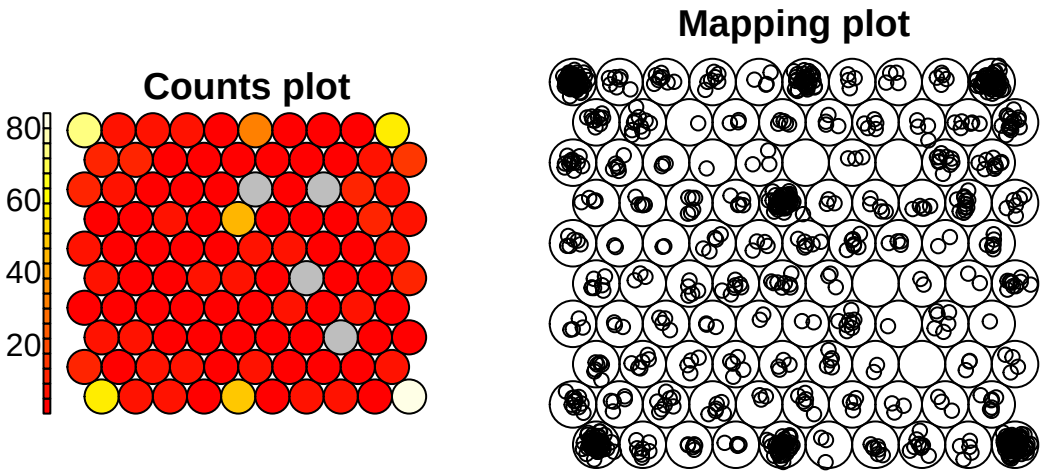
This type of chart allows to establish the role of variables in the definition of the different areas that comprise the topological map. This is important for the interpretation of the results. This chart represents the vector of weights in a pie chart for each cell of the map.



We note that the topics are well distinguished, and that in the center of the map we can found the mixed-topic document that are those characterized by 2 or more different topics.

Count Plot

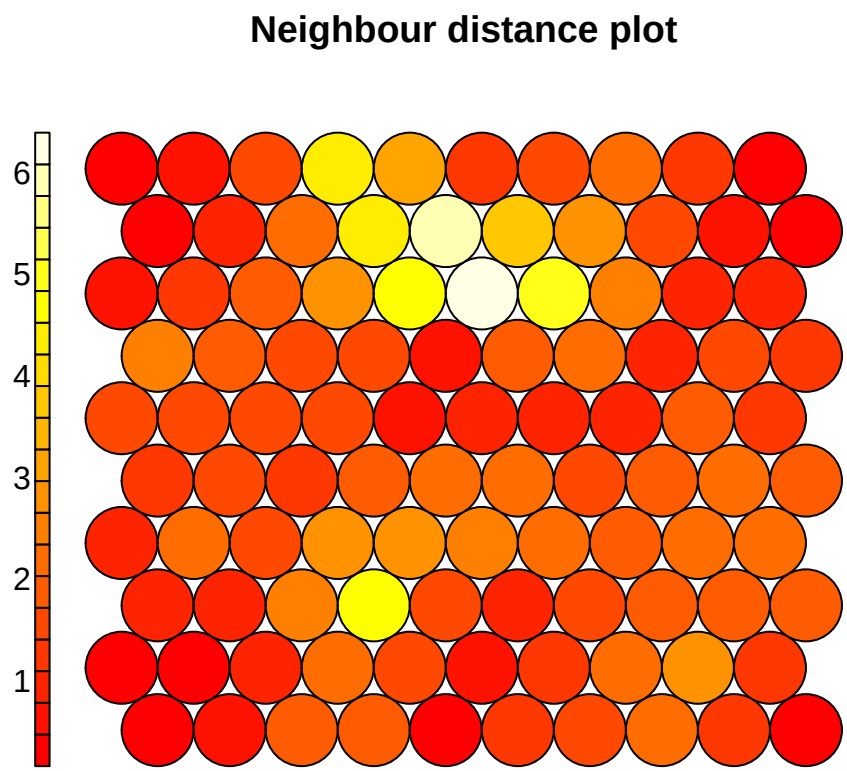
Count plot show how many artcles are in each part of kohonen map, we wish to have all the part filled up. We can then identify high-density areas. Ideally, the distribution should be homogeneous. The size of the map should be reduced if there are many empty cells. Conversely, we must increase it if areas of very high density appear.



In this case we only have 4/100 empty point so we decide that the dimension is fine.

Neighbour distance plot

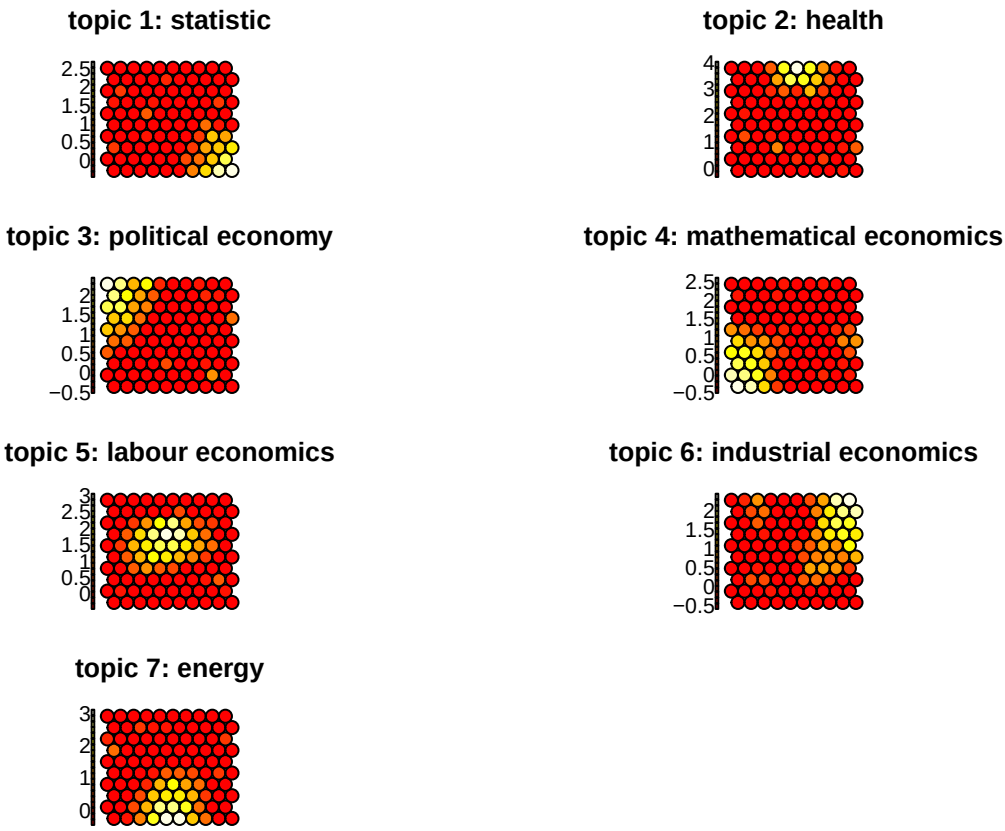
Neighbour distance plot. Called "U-Matrix" (unified distance matrix), it represents a selforganizing map (SOM) where the Euclidean distance between the codebook vectors of neighboring neurons is depicted in a range of colors. According to the package documentation, the nodes that form the same group tend to be close. Border areas are bounded by nodes that are far from each other.



In our chart , the nodes which are close to the others are red-colored. We observe that we have an ovest part in which documents seems to be more distanced.

Kohonen map fot each topic

Rather than making a single chart for all the variables, we can make a graph for each variable, trying to highlight the contrasts between the high and low value areas. This univariate description is easier to understand.



GENERAING DATASET FOR VISUALIZATION ON TABLEAU

Since package Kohonen doesn't allow to create an interactive visualization we need to extract from the som_model some information about coordinates of each point.

We then need to attach information about topic type for each document. (choosing topic with higher gamma probabilities)

We use as distances jittered values to better separate values.

This plot is a traslated version of kohonen map on ggplot. This show what we'll obtain in tableau.

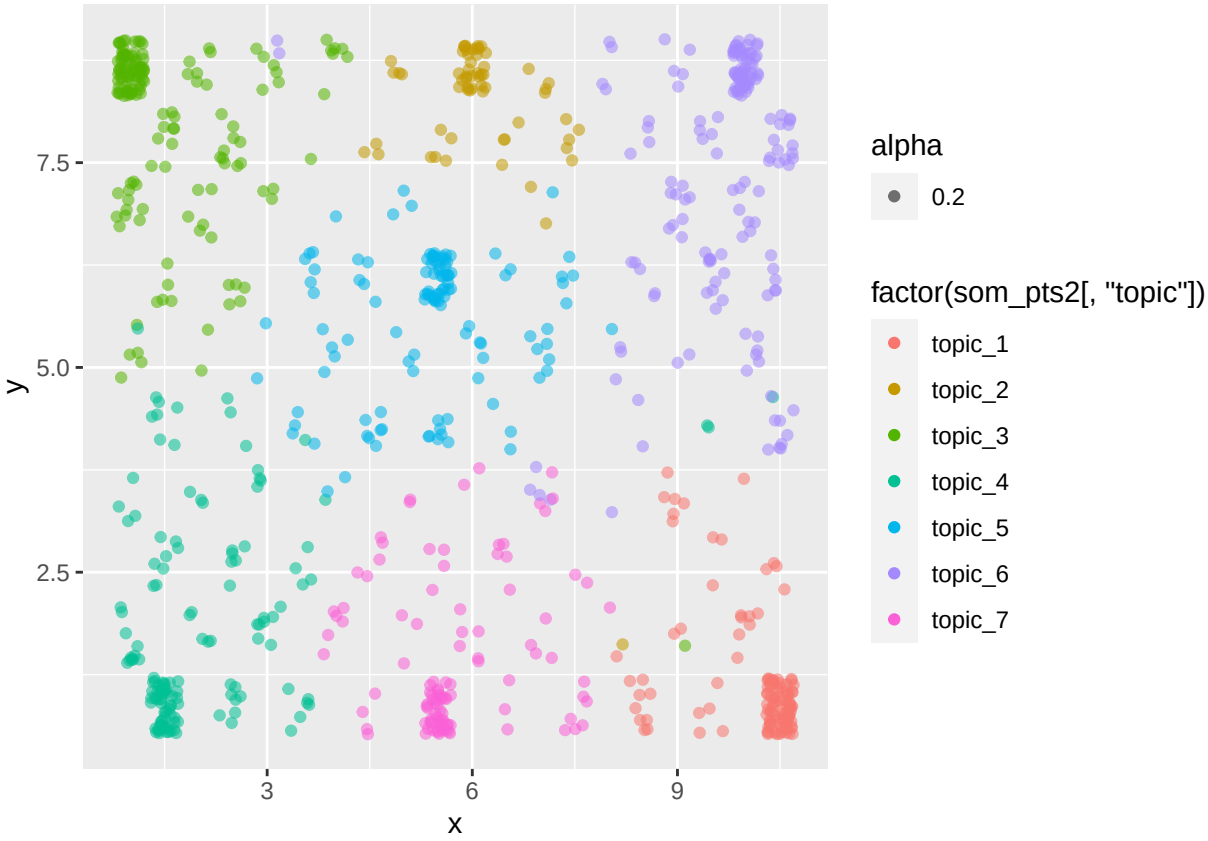


Tableau visualization is available at the following link: https://public.tableau.com/profile/alessandro5441#!/vizhome/DocumentsandAuthorsmapping-LDA_16122629406460/Foglio1?publish=yes

CONCLUSIONS

Topic modeling analysis performed did not have excellent results in terms of consistency of the topics identified, however based on our judgment the classification seems correct. The advantage of this model is that is really fast, in fact LDA takes few seconds to run on almost a 1000 abstract. The negative aspect is that it does not take in account the semantic. Future developement could take in account also this aspect to obtain more specifc results.