Hierarchic Topic Models Visualization

Dmitriy S. Fedoriaka[1]

[1]MIPT

Abstract

Hierarchic topic models are good tool for representing big amount of text documents. However, displaying such models on screen is difficult problem itself.

This paper discusses problem of hierarchic topic models visualization. It introduces concept of tree visualization with polygons. Also it considers problems of quality measuring and representation of additional information.

This article also describes implementation of visualization algorithms with usage of BigARTM topic modeling library.

Keywords: *Topic Modeling, Visualization, BigARTM.*

1 Introduction

Topic models are good tool for finding hidden structure of big set of documents. Well-built topic model can help users to explore big sets of documents (web sites, blog entries, scientific articles, books).

But very big data sets can't be satisfactory described by flat (i.e. one-layer) topic models — each topic will contain too many documents. That's why we need to use hierarchical (i.e. multi-layer) topic models. But it's non-trivial problem to represent tree of hierarchical topic model.

This paper describes approach for building and visualization of hierarchical topic models. It discusses problem statement, building topic model, extracting tree from it and visualizing this tree with polygons.

2 General problem statement

Given set of documents (each document is set of words). We need to build such tree, that documents are its leafs and inner nodes corresponds to topics. Here topics are understood informally: documents belongs to the same topic, if they are about similar things.

Next problem is to show that tree in such way, that makes it easy to user to observe whole set of documents and effectively perform search tasks.

3 Mathematical problem statement

Below problem of building hierarchic topic models will be formulated, as it is formulated in BigARTM (open-source library for topic modelling).

First, let's consider documents as set of words (also it's called "bag of words"). Let's define documents as elements of set W, which will be referred to as dictionary. Let's denote D — set of all documents. Now we can count $F_{w,d}$ — frequency of word w in document d. It will be equal to $\frac{|d|_w}{|d|}$, where $|d|$ is number of words in document d and $|d|_w$ is number of occurrences of word w in that document. Now we have stochastic matrix F of dimension $|W| \times |D|$, where $F_{w,d} = p(w|d)$ is distribution over set W in document d.

Now let's introduce topics also as distributions over dictionary. Let's denote T — set of topics (in this model we consider number of topics to be known). Then, topics can be given by matrix Φ of dimension $|W| \times |T|$, where $F_{w,t}$ is probability of occurrences of word w in topic t.

In general, each document d can be related to several topics. Let's denote $\Theta_{t,d} = p(t|d)$ — distribution over topics in documents d.

Let's assume that each word relates to one topic (however, different occurrences of the same word can relate to different topics). Than if word w occured ifn document n_i times relating to topic t_i, then total number of occurrences will be $\sum n_i$. So, we can write

$$F_{w,d} = \sum_{t \in |T|} \Phi_{w,t} \cdot \Theta_{t,d}$$

or simply

$$F = \Phi\Theta.$$

So, we have problem of matrix decomposition. It can be formulated as problem of likelihood maximization:

$$\sum_{d \in D} \sum_{w \in W} |d|_w \ln \sum_{t \in T} \Phi_{w,t}\Theta_{t,d} \to \max \tag{3.1}$$

and effectively solved by EM-algorithm.

That's exactly what BigARTM does. For more information refer to [2] and [3].

Now let's formulate problem statement for hierarchic topic model. Hierarchic topic model is set $L = L_1, \dots, L_{|L|}$ of levels. Those level consists of topics. Topic of level L_i are distributions over words for topics of level L_{i+1} considered as documents. Topics of level $L_|L|$ are distributions for documents, as before.

Level L_1 will be referred to as top level, and level $L_{|L|}$ - as bottom level.

Building of hierarchical topic structure in BigARTM goes from top to bottom. So, it is set of L problems (3.1). For statement of each problem refer to [5].

4 Building of hierarchical tree

Assume we have simple topic model ($L = 1$). BigARTM model, having been fit, provides us with stochastic matrix Θ, where $\Theta_{t,d} = p(t|d)$ is topic distribution over document d. The obvious way to assign topic for document d is get the most relevant topic:

$$Topic(d) = \arg\max_t \Theta_{t,d}. \tag{4.1}$$

If we have hierarchical topic model (i.e. with more then one level), we have $L-1$ such matrices Θ.

So, we create nodes for root of the tree, for each topic of each level and for each document. Then for we assign topics from level L for each document using (4.1). Then for each topic of level $i = 2, \dots, L$ we assign it's parent topic from level $i-1$ using (4.1) fr corresponding matrix Θ. Finally, parent topic of each topic of first level is root.

5 Dealing with "multitopical" documents

There are document which can be attributed to more then one topic (e.g. interdisciplinary articles). Great feature of BigARTM is that it can recognize such documents d: for them there will be several such topics t that $\Theta(t, d)$ are more greater then another values in d-th row of $\Theta(t, d)$.

There are two possible ways to show information about multiple topics in visualization. First way is to make several copies of node, representing the document and put them as children to each topic, which can be parent to this document.

Formally, let's determine significance level ε and define

$$MultiTopic(d) = \arg\max_t \Theta_{t,d} \cup \{t | \Theta_{t,d} > \varepsilon\}. \tag{5.1}$$

Then let's put document d as child in all topics from set $MultiTopic(d)$.

There can also be also "multitopical" topics. We can deal with them in similar way, only we need to copy nodes with their sub-trees.

The second way is to not change tree structure, but arrange topic representations in such way that if document relates to topics A and B, it will be showed on intersection of topics A and B. It is especially good for topics of lower levels. We can show that this topic is interdisciplinary area between topics of higher level. This approach will not be discussed in the paper.

6 Visualization of hierarchical tree

We want to implement principle **"Overview first, zoom and filter, details on demand"**, stated in [1].

Partition of documents into different topics is partition of one set into smaller parts. The most intuitive way to represent it using two-dimensional plane is to divide figure into smaller parts.

So, let's represent documents with polygons (because any connected figure can be approximated by polygon).

Now let's formalize problem of representation. Given tree $G = (V, E)$. Representation of that tree is such function f, that defines for each vertex V polygon $f(V)$ such as:

1. If vertex v has children (denote them as set C), then

$$f(v) = \bigcup_{c \in C} f(c) \tag{6.1}$$

2. If c_1, c_2 — children of the same vertex v, then

$$\mu(f(c_1) \cap f(c_2) = 0 \tag{6.2}$$

Here μ means square.

Figure 1 describes this approach.

Now let's discuss algorithms of building such representation.

First approach is to use Voronoi diagrams. The advantage of that approach is that all polygons will be convex. Root polygon should be given. Then we partition it into polygons. The simplest way to do it is to throw random points for each children, then build Voronoi diagram for them. Some of cells will be infinite and we have to intersect them with root polygon.

Then the same routine should be done for each of obtained polygons and so on. Figure 2a displays result of such algorithm.

There is system FoamTree which implements tree visualizations with Voronoi diagrams. It is able to build well-looking partitions, so most of polygons have 5 or 6 angles. Figure 2c displays visualization of topic model with usage of FoamTree. For more information about FoamTree, refer to [4].

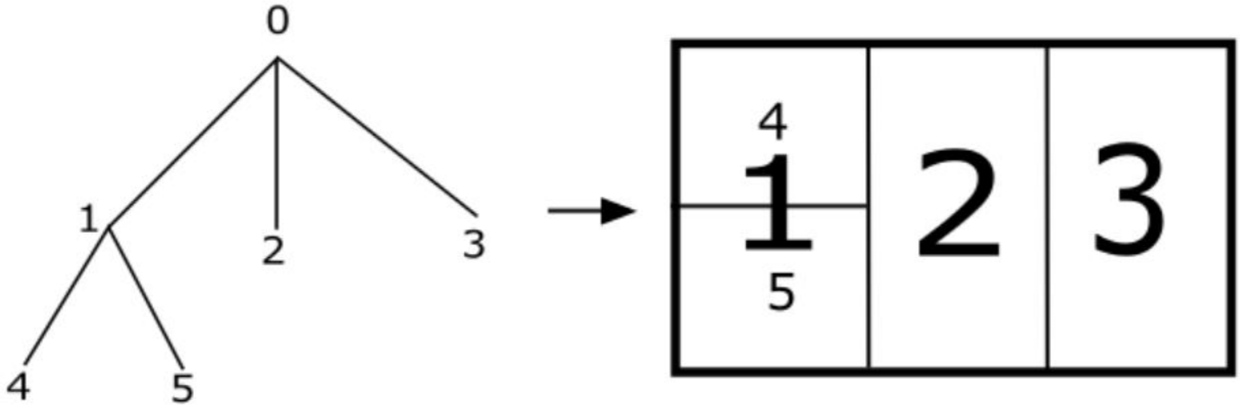


Figure 1: Tree representation with inserted polygons

7 Grid visualization

Another approach is to use fixed grid and try to put documents in cells. The advantage of this approach is that representation of documents will be of the same simple shape and area can be used more effectively.

Having arranged documents, we can build polygons for topics from bottom to top, uniting polygons. Formally, conditions (6.1) and (6.2) will be satisfied but representations for topics may not be connected figures. So, we can formulate problem as finding such arrangement, that all unions of cells corresponding to the same topics are connected figures. It isn't clear, how to effectively solve this problem.

But this problem can be reduced to optimization task. For simplicity, let's assume that we have only one layer of N topics, which containing S_i topics. Let's represent square grid as matrix M of size $W \times H$, such as $\sum_{i=1}^{N} S_i = W \cdot H$. $M[x,y] = 0$ if cell (x,y) is empty and $M[x,y] = k$ if cell (x,y) is occupied by document of topic k.

So, we have restriction

$$\sum_{x=1}^{W}\sum_{y=1}^{H}[M[x,y] = i] = S_i, \quad \forall i \in \overline{1,N} \tag{7.1}$$

Let's define energy as follows

$$U = \sum_{(x_1,y_1)\neq(x_2,y_2)} \rho((x_1,y_1);(x_2,y_2)) \cdot A_{M_{x_1,y_1},M_{x_2,y_2}}$$

where

$$A[i,j] = \begin{cases} 1, \text{if } i = j > 0 \\ 0, \text{otherwise} \end{cases}$$

$$\rho((x_1,y_1);(x_2,y_2)) = \sqrt{((x_1-x_2)^2 + (y_1-y_2)^2)}$$

Now we have optimization problem $U \to \min$ such that (7.1) is satisfied.

Now let's show how this problem can be reduced to problem of quadratic programming. Consider 3-dimensional array Z of dimension $W \times H \times N$, defined as follows:

$$Z[x,y,i] = \begin{cases} 1, \text{if } M[x,y] = i \\ 0, \text{if } M[x,y] \neq i \end{cases}$$

Then U will be written as follows:

$$U = \sum_{x_1=1}^{W} \sum_{y_1=1}^{H} \sum_{x_2=1}^{W} \sum_{y_2=1}^{H} \rho(x_1, y_1; x_2, y_2) \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} A[i_1, i_2] \cdot Z[x_1, y_1, i_1] \cdot Z[x_2, y_2, i_2]$$

Condition on total square of i-th area:

$$\sum_{x=1}^{W} \sum_{y=1}^{H} Z[x, y, i] = S_i$$

Besides that, each cell $M[x, y]$ can be painted not more than in one color:

$$\sum_{i=1}^{N} Z[x, y, i] \leq 1$$

So we have problem of discrete optimization:

$$\boxed{\sum_{x_1=1}^{W} \sum_{y_1=1}^{H} \sum_{x_2=1}^{W} \sum_{y_2=1}^{H} \rho(x_1, y_1; x_2, y_2) \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} A[i_1, i_2] \cdot Z[x_1, y_1, i_1] \cdot Z[x_2, y_2, i_2] \to \min_{Z}}$$

Such that

$$\begin{cases} \forall i \in \overline{1, N} \quad \sum_{x=1}^{W} \sum_{y=1}^{H} Z[x, y, i] = S_i \\ \forall (x, y) \in \overline{1, W} \times \overline{1, H} \quad \sum_{i=1}^{N} Z[x, y, i] \leq 1 \end{cases}$$

Advance of this approach that modifying function $A[i, j]$ we can display hierarchical topic models and even consider interconnextion between topics in one level and show close by content topics with geometrical close polygons.

Unfortunately, this problem can be solved only approximately with randomized methods, which work very long at big sets, so method is almost useless on practice.

Figure 2b displays example of such visualization.

8 Interactive display

Now we have tree of hierarchy displayed as set of inserted polygons. We cannot change coordinates of polygons. All what we can do is zooming in and out and moving the picture. However, having used those actions, we can display more information then with static picture.

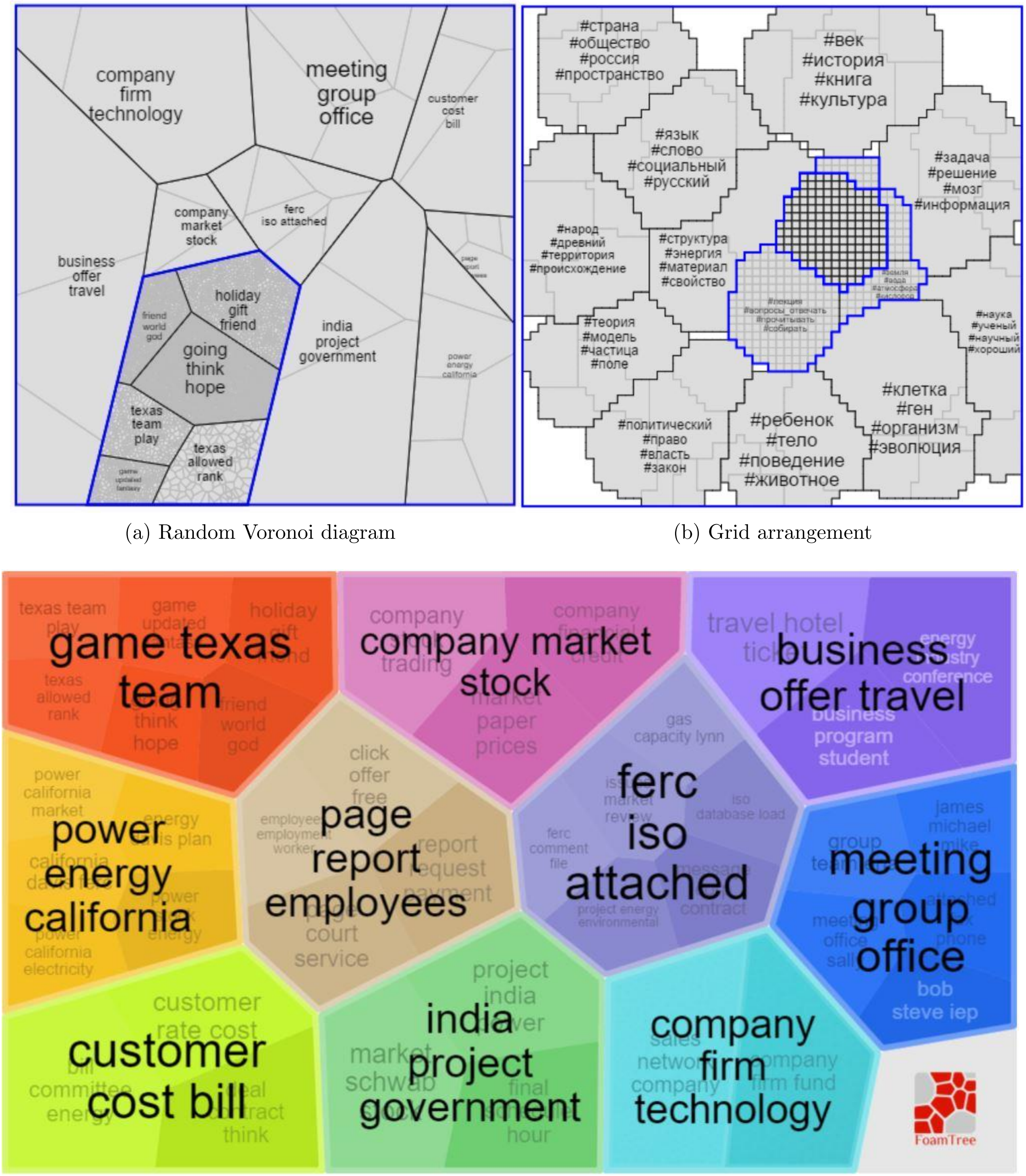
Now let's define minimal functionality of interactive display, which will implement principle **"Overview first, zoom and filter, details on demand"**.

1. **Zooming in and out**. Initially we display all set of documents as a single polygon (i.e. polygon, corresponing to node of tree). Then user should be able to zoom in some area and see topics of lower levels and documents. After that he may want to explore another topic, so he have to be able to zoom out.

2. **Parallel translation**.

3. **Text in polygons.** It can be titles for documents and most relevant words for topics.

4. **Clickable polygons.** If we will show text in all polygons, text will overlap and user will not be able to read it. So, if we show title of topic, we cannot show titles of subtopics or documents. But if user wants to explore topic, he clicks on the corresponding polygon and sees structure of the topic. When user clicks on polygons of documents, he may be redirected to page, containing this document. So, such visualization can be used as interactive map of site.



(a) Random Voronoi diagram

(b) Grid arrangement

(c) FoamTree

Figure 2: Visualizations

5. **Showing information when cursor hovers below polygon.** It can be snippet (i.e. short description) of documents. It will be much faster for user to read information without clicking that click, read document and return back.

This concept of visualization is developed mostly for web pages and all enlisted features can be easily implemented using JavaScript. FoamTree has all these features.

9 Topic naming

It's very important to give to topics short names, which will provide information about content.

Now creating of universal algorithm for topic naming is an open problem. Some approaches are discussed in [7, 8].

In this project we use trivial approach — taking *top words*.

For topic t top words are such set $TW(t) \subseteq D$, that

$$\forall d \in TW(t), \tilde{d} \in D \setminus TW(t) \quad \Theta_{d,t} \geq \Theta_{\tilde{d},t}.$$

In other words, they are the most encountered words of topic t. If *stop-words* (common words, which are encountered in almost each text, like grammatical auxiliary words) were excluded from documents in advance, top words are likely to be the most relevant words for topic description. Having taken 3 or 4 of them, we receive acceptable label for the topic.

10 Quality measurement

There are some quantitative metrics for topic models (such as perplexity), but there aren't any metrics which can describe, how good visualization is.

The main goal of visualization is to help a human being to explore rage set of documents. That's why appropriate metrics is time, spent by user for accomplish certain task.

The task can be, for example, finding document when user knows what it is about, or finding document similar to given. Time can be measured in real time, or in number of clicks.

This metric should be compared to similar metric for common search systems (e.g. simple search for words in documents). Let's refer to such system as to *basic system* and to our system as to *target system*.

So, we can define "improvement coefficient" as

$$\alpha = \frac{t_0}{t},$$

where t_0 is time spent by user to accomplish task using basic system and t is time spent by user on certain task using basic system and t_0 — by using target system. For purity of the experiment we should give the same task for two disjoint groups of assessors. First group should use basic system, another — target system. Then we are to average results and calculate α. For better accuracy, such experiment should be repeated for different tasks and data sets.

We can claim that visualization provided by target system makes sense, if $\alpha > 1$. It's possible that for some problems or data sets will be $\alpha \lesssim 1$. That would mean that target system isn't appropriate for that problems or data sets.

11 Implementation

During this work a web application for topic models was created. It uses BigARTM for building of topic models and FoamTree for representing of tree with Voronoi diagram. This application was tested on several UCI datasets: "kos", "enron" and "nips" from [6] and sets of documents from site postnauka.ru.

References

[1] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. *Proceedings of the IEEE Symposium on Visual Languages*. 1996.

[2] K. Vorontsov, A.Potapenko. *Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization.* http://www.machinelearning.ru/wiki/images/1/1f/Voron14aist.pdf

[3] Online documentation for BigARTM. http://docs.bigartm.org/en/stable/index.html

[4] Online documentation for FoamTree.
https://get.carrotsearch.com/foamtree/latest/api/index.html

[5] N. A. Chirkova, K. V. Vorontsov. Additive Regularization for Hierarchical Multimodal Topic Modelling. JMLDA, vol.2, #2. 2016.

[6] UCI datasets. https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

[7] Jey Han Lau, Karl Grieser, David Newman, Timothy Baldwin. Automatic Labeling of Topic Models. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Portland, Oregon. 2011.

[8] Kou Wanqiu, Li Fang, Timothy Baldwin. Automatic Labeling of Topic Models using Word Vectors and Letter Trigram Vectors. *Proceedings of the Eleventh Asian Information Retrieval Societies Conference (AIRS 2015)*. Brisbane, Australia. 2015.