Hierarchical Thematic Classification of Major Conference Proceedings

Arsentii Kuzmin[1 [0000−0002−4636−6804]], Alexander Aduenko[1 [0000−0002−0959−2052]], and Vadim Strijov[1 [0000−0002−2194−8859]]

Moscow Institute of Physics and Technology, Inststitutskii per. 9, 141700, Dolgoprudny, Russia
{arsentii.kuzmin, aduenko, strijov}@phystech.edu
https://mipt.ru/en/

Abstract. In this paper we develop a decision support system for hierarchical text classification. We consider text collections with fixed hierarchical structure of topics given by experts in the form of a tree. The system sorts the topics by relevance to a given document. The experts choose one of the most relevant topics to finish the classification. We propose a weighted hierarchical similarity function to calculate topic relevance. The function calculates the similarity of a document and a tree branch. The weights in this function determine word importance. We use the entropy of words to estimate the weights.
The proposed hierarchical similarity function formulate a joint hierarchical thematic classification probability model of the document topics, parameters, and hyperparameters. The variational Bayesian inference gives a closed form EM algorithm. The EM algorithm estimates the parameters and calculates the probability of a topic for a given document. Compared to hierarchical multiclass SVM, hierarchical PLSA with adaptive regularization, and hierarchical naive Bayes, the weighted hierarchical similarity function achieves superior ranking accuracy on a collection of abstracts from the major conference EURO and a collection of websites of industrial companies.

Keywords: Text classification · Bayesian variational inference · Document similarity · Word entropy · Hierarchical categorization · Text ranking.

1 Introduction

A thematic model of a text collection is a map, which determines a set of topics from a given hierarchical structure of topics for each document from the collection. The text collections are scientific abstracts [17, 3], conference proceedings, text messages from social networks [26], web sites [28], patent descriptions, and news articles [15, 11]. The thematic model assists in searching through collections efficiently. However the model construction is often labour intensive. Some collections already have a structure and a subset of documents that have been partly classified by experts. To simplify the procedure of expert classification the authors propose an algorithm that ranks a collection's topics for a given document. On user demand, the algorithm puts a new document into the topic with the highest rank.

This paper investigates the thematic modelling problem for partially labelled collections with fixed expert tree structure of topics [22, 18, 12]. In the tree structure, the leaf topic of a document determines the topics for this document on the other levels of the hierarchy. Thus, the required solution is a map, which determines the ranks of the leaf topics for a given document. The ranks of the expert topics determine the quality of the solution.

⋆ Supported by the Government of the Russian Federation (agreement 05.Y09.21.0018) and the RFBR (project 19-07-0875). This paper contains results of the project Statistical methods of machine learning, which is carried out within the framework of the Program "Center of Big Data Storage and Analysis" of the National Technology Initiative Competence Center. It is supported by the Ministry of Science and Higher Education of the Russian Federation according to the agreement between the M.V. Lomonosov Moscow State University and the Foundation of project support of the National Technology Initiative from 11.12.2018, No 13/1251/2018.

The relevance of a leaf topic to a given document determines the rank of the topic. The relevance is determined by the value of a discriminant function or the probability estimate of a discriminative or generative model. For a non-hierarchical classification problem [16, 17, 21], SVM, kNN, and Neural Networks return a value of the discriminant function. In [7, 8, 13], the Naive Bayes, Multinomial logistic regression, and dPM models give probability estimations for a topic of a given document.

In hierarchical classification the "top-down" approach [12] yields better results than non-hierarchical classification among leaf-level topics. Starting from the top of the hierarchy it relates a document to one of the children topics using non-hierarchical classification methods [24, 28]. Its drawback is that a misclassification at the top level immediately leads to a misclassification on the leaf level. To rank the leaf topics using a "top-down" approach the algorithm sorts them according to the ranks of the parent topics on each level. An alternative method is to consider all clusters of the tree branch at once [22, 27].

Taking into account the word importance improves the classification quality. A common approach is to change a frequency-based document description to $\mathrm{tf} \cdot \mathrm{idf}$ features [25]. Another approach is to optimize a weighted metric or similarity function [20, 4, 29, 23]. The disadvantage of the last approach is the huge number of optimization parameters, equalling the dictionary size of the collection. In [19], the authors use the entropy to estimate the importance of words and reduce the number of parameters. In this paper we improve this approach and generalize it to the hierarchical case.

We propose a weighted hierarchical similarity function of a document and a branch of a cluster tree. The function considers the word importance and hierarchical structure of the collection. We put priors on parameters of the similarity function to regularize them and take into account our assumptions regarding their default values. The similarity function formulate a joint hierarchical thematic classification probability model of document topics, parameters, and hyperparameters. We use the variational Bayesian inference [6, 9, 5] to derive the EM algorithm and estimate the probabilities of topics for unlabelled documents.

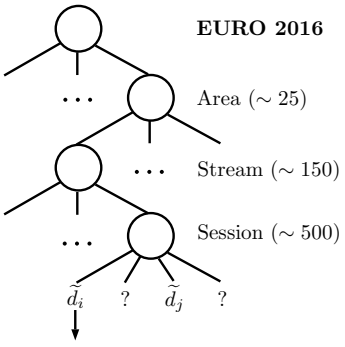


Fig. 1: Structure of the EURO conference.

We consider the process of constructing a thematic model of the major conference "European Conference on Operational Research (EURO)" as an example of the thematic modelling task. A program committee builds the thematic model for this conference from a set of received abstracts every year. The structure of this model consists of 26 major Areas, each Area consists of $10 - 15$ Streams, each stream consists of $5 - 10$ Sessions, and each Session consists of four talks 1. Participants send short abstracts to program committee to apply. There are two types of participants: invited participants and new participants. The invited participants already have a determined session, so the collection of abstracts is partly labelled. For each new participant, the program committee should choose the most relevant session according to his/her abstract and the conference structure. The program committee invites up to 200 experts from different research areas to construct the thematic model.

We construct a decision support system for the creation of the thematic model of the conference, which gives the expert a ranked list of possible clusters for a given document. We use expert models of this conference from previous years to estimate the parameters, and we compare the quality of the proposed algorithm with commonly used classification methods.

2 Weighted hierarchical similarity function

Denote a word as w be. A document d is an unordered set of words $\{w_1, w_2, \ldots, w_{|d|}\}$. A document collection D is an unordered set of documents

$$D = \{d_1, d_2, \ldots, d_{|D|}\}. \tag{1}$$

A dictionary W of the collection D is an ordered set of unique words w that form the collection D. Each document d_n is represented by a real value vector $\mathbf{x}_n$. The element $x_{i,n}$ of $\mathbf{x}_n$ equals the number of words w_i in the document d_n.

A cluster c is a subset of documents from the collection D. The experts define a collection structure as a graph of topics. In this paper we consider only trees as possible collection structures. Each node (leaf) of the topic tree corresponds to a cluster c of documents from this topic. Let h be the height of the tree. Indexes ℓ and k of a cluster $c_{\ell,k}$ denote the level in the tree and the index on this level, respectively. Cluster $c_{1,1}$ is a root of the tree. Let K_ℓ be the number of clusters on the level ℓ. Cluster c_1 is a parent cluster for c_2 if it contains all documents d from c_2. Then cluster c_2 is a child cluster of c_1. Let B be an operator that returns that parent cluster of a given cluster. We use the B operator $h - \ell$ times $B^{h-\ell}(c_{h,k})$ to get the parent cluster on the level ℓ of the lowest level cluster $c_{h,k}$ (see. Fig. 2).

Let $c(d)$ be an expert cluster of document d on the lowest level h. Matrix $\mathbf{Z}$ determines the expert clusters on the lowest level for all documents:

$$z_{nk} = [d_n \in c_{h,k}], \quad z_{nk} \text{ is the element of } \mathbf{Z}. \tag{2}$$

The leaf cluster determines the classification of the document on all other levels of the hierarchy. Thus, the matrix $\mathbf{Z}$ determines an entire expert thematic model.

Quality criterion for hierarchical ranking. For each document the algorithm ranks all leaf clusters according to their relevance. Then, the expert chooses one best cluster from the ranked set. The rank of the chosen cluster determines the quality of the ranking: the lower the rank, the better the quality.

Let S^{K_h} be a permutation set of order K_h. Let R be a relevance operator. The relevance operator maps each document $\mathbf{x} \in \mathbb{R}^{|W|}$ to a cluster permutation $q(\mathbf{x}) \in S^{K_h}$ of level h. The

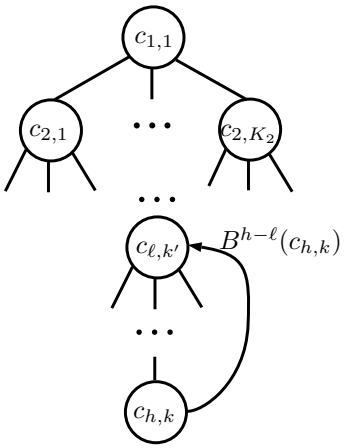


Fig. 2: Basic notation in the hierarchical structure of the collection.

clusters in permutation $q(\mathbf{x})$ are sorted by relevance to document $\mathbf{x}$ in descending order. The rank of each cluster is equal to its position in the permutation. The goal is to find the operator $R(\mathbf{x})$ that has the best quality on the expert classification $\mathbf{Z}$:

$$\text{AUCH}(R, D, \mathbf{Z}) \to \max,$$

where $\text{AUCH}(R, D, \mathbf{Z})$ is a quality function that depends on the ranks of the expert clusters.

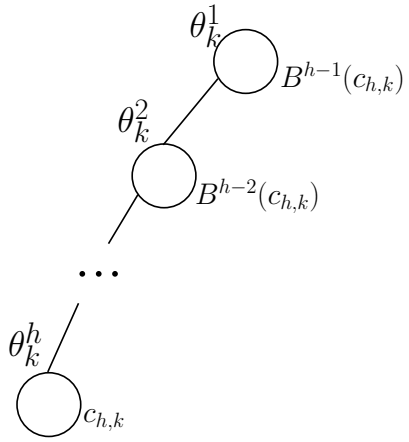


Fig. 3: A branch with the number k of a cluster hierarchy. The value of θ_k^ℓ denotes the weight of the cluster $c_{\ell,k}$ in the branch.

A common ranking quality criterion is the discounted cumulative gain DCG_k and $p@k$:

$$\text{DCG}_k = \text{rel}_1 + \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2 i}, \quad p@k = \frac{r_k}{k}, \tag{3}$$

where rel_i is the relevance of the cluster with rank i, and r_i is the number of relevant clusters among the first i clusters in the permutation $q(\mathbf{x})$. In our case, an expert always selects a single cluster. If he selects the cluster with the rank j, then $\text{rel}_i = 1$ if $i = j$ and 0 otherwise. The same is true for r_k: if the rank of the selected cluster is less then k, then $r_k = 1$ and 0 otherwise.

DCG_k and $p@k$ are too detailed in the context of our problem. We propose a simplified quality criterion AUCH instead. By $\text{pos}\big(R(\mathbf{x}_n),\, c(\mathbf{x}_n)\big)$ denotes the rank of the expert cluster $c(\mathbf{x}_n)$ of document $\mathbf{x}_n$ according to the permutation $R(\mathbf{x}_n)$. We introduce a monotone increasing cumulative histogram of documents with respect to their expert ranks:

$$\#\{n:\ \text{pos}\big(R(\mathbf{x}_n),\, c(\mathbf{x}_n)\big) \le k\}, \quad k \in [1,\ K_h]. \tag{4}$$

The quality criterion $\text{AUCH}(R) \in [0,\, 1]$ (from Area Under Cumulative Histogram) equals the area under this histogram, normalized by the number of documents and clusters:

$$\text{AUCH}(R) = \frac{1}{K_h|D|} \sum_{k=1}^{K_h} \#\{n:\ \text{pos}\big(R(\mathbf{x}_n),\, c(\mathbf{x}_n)\big) \le k\}. \tag{5}$$

Fig. 8 illustrates an example of the envelope curve for the histogram. The value $\text{AUCH}(R) = 1$ corresponds to the optimal case, when the expert cluster of each document is located in the first position in the permutation.

Weighted similarity of a document and cluster. To rank the clusters one should estimate the relevance value of each cluster $c_{\ell,k}$ to a given document $\mathbf{x}$. Let the similarity $s(\mathbf{x},\, \mathbf{y})$ of

documents $\mathbf{x}$ and $\mathbf{y}$ be a weighted cosine similarity function

$$s(\mathbf{x},\, \mathbf{y}) = \frac{\mathbf{x}^\intercal \mathbf{\Lambda} \mathbf{y}}{\sqrt{\mathbf{x}^\intercal \mathbf{\Lambda} \mathbf{x}}\sqrt{\mathbf{y}^\intercal \mathbf{\Lambda} \mathbf{y}}}. \tag{6}$$

If the denominator equals zero, the similarity also equals zero. A symmetric non-negative definite matrix $\mathbf{\Lambda}$ determines the importance of the words. In this paper we consider a diagonal $\mathbf{\Lambda}$, because optimization of all its $|W| \times |W|$ elements leads to an inadequate increase in model complexity. We normalize all documents to unclutter the notation:

$$\mathbf{x} \mapsto \frac{\mathbf{x}}{\sqrt{\mathbf{x}^\intercal \mathbf{\Lambda} \mathbf{x}}}, \quad s(\mathbf{x},\, \mathbf{y}) = \mathbf{x}^\intercal \mathbf{\Lambda} \mathbf{y}. \tag{7}$$

Let $\boldsymbol{\mu}(c_{\ell,k})$ be the mean vector of a cluster $c_{\ell,k}$

$$\boldsymbol{\mu}(c_{\ell,k}) = \frac{1}{|c_{\ell,k}|} \sum_{\mathbf{x} \in c_{\ell,k}} \mathbf{x}. \tag{8}$$

The similarity $s(\mathbf{x},\, c_{\ell,k})$ of a document and a cluster is the similarity function value (6) of the document vector $\mathbf{x}$ and the mean vector $\boldsymbol{\mu}(c_{\ell,k})$ of the cluster:

$$s(\mathbf{x},\, c_{\ell,k}) = \mathbf{x}^\intercal \mathbf{\Lambda} \boldsymbol{\mu}(c_{\ell,k}). \tag{9}$$

Entropy model of word importance. The number of weight parameters in the diagonal matrix $\mathbf{\Lambda}$ equals the size of the dictionary $|W|$. We propose an entropy model to decrease the number of parameters and avoid overfitting. It maps the entropy of a word w_m to the importance λ_m of this word.

Words that separate clusters are the most important for classification. To understand what it means for a word to separate clusters, consider the following example. Let all documents from a cluster $c_{\ell,k}$ contain a word w, while all documents from the other clusters do not contain w. Then it is adequate to classify a new unlabelled document with the word w to the class $c_{\ell,k}$. The entropy approach formalizes this idea.

Let $p^\ell_{m,k} = p(c_{\ell,k}|w_m)$ be a probability of cluster $c_{\ell,k}$ given word w_m. We can estimate $p^\ell_{m,k}$

$$\mathbf{p}^\ell_m = \big[\mu(c_{\ell,1})_m,\, \ldots,\, \mu(c_{\ell,K_\ell})_m\big]^\intercal, \quad \mathbf{p}^\ell_m \mapsto \frac{\mathbf{p}^\ell_m}{\|\mathbf{p}^\ell_m\|_1}, \tag{10}$$

where $\boldsymbol{\mu}(c_{\ell,k})_m$ is the m-th component of the cluster's $c_{\ell,k}$ mean vector. We define the entropy of word w_m according to expert classification on level ℓ as

$$\mathsf{H}^\ell(w_m) = -\sum_{k=1}^{K_\ell} p^\ell_{m,k} \log(p^\ell_{m,k}). \tag{11}$$

The smallest entropy value $\mathsf{H}^\ell(w_m) = 0$ corresponds to the case, in which the word w_m occurs only in the documents of one cluster and this cluster is separate from the others. The maximum entropy value corresponds to the uniform distribution of the word w_m over all clusters, $p^\ell_{m,k} = \text{const}$. In this case, w_m is an unimportant word.

In the case of a hierarchical structure, we calculate the entropy of the word according to each level ℓ. We combine these values to obtain the importance value λ_m of the word w_m:

$$\lambda_m = 1 + \sum_{\ell=1}^{h} \alpha_\ell \log\big(1 + \mathsf{H}^\ell(w_m)\big). \tag{12}$$

Parameter α_ℓ determines the influence of word entropy on level ℓ to the importance of the words. The expression $\log\big(1 + \mathsf{H}^\ell(w_m)\big)$ does not contain any variables, so we calculate it for each word and level ℓ and denote

$$\iota_{m\ell} \equiv \log\big(1 + \mathsf{H}^\ell(w_m)\big).$$

Then, model (12) takes the form

$$\lambda_m = 1 + \boldsymbol{\alpha}^\intercal \boldsymbol{\iota}_m. \tag{13}$$

Hierarchical ranking. One solution for hierarchical ranking is the top-down approach. Let $C_h(c_{\ell,k})$ be a set of level h clusters, which are children clusters for $c_{\ell,k}$. Let $\mathrm{idx}(c_{\ell,k})$ be the rank of the cluster $c_{\ell,k}$ on level ℓ. We go down from the root of the tree and, on each level ℓ, rearrange the lowest-level clusters $C_h(c_{\ell,k})$ so as to preserve the condition that

$$\mathrm{idx}(c_{\ell,k_1}) < \mathrm{idx}(c_{\ell,k_2}) \Rightarrow \mathrm{idx}(c_{h,k_1'}) < \mathrm{idx}(c_{h,k_2'}), \quad \forall k_1,\ k_2,\ c_{h,k_1'} \in C_h(c_{\ell,k_1}),\ c_{h,k_2'} \in C_h(c_{\ell,k_2}).$$

This approach retains the top-down disadvantage: an incorrect ranking on the high level of the tree immediately leads to incorrect ranking on the lowest level. Let i be the rank of the cluster $c_{2,\hat{k}}$ that contains expert cluster $c(\mathbf{x})$ for a given document. Then, the rank of the expert cluster for this document on the lowest level h will be at least

$$\mathrm{idx}(c(\mathbf{x})) > \sum_{k\,:\,\mathrm{idx}(c_{2,k})<i} |C_h(c_{2,k})|.$$

We propose the hierarchical similarity function to address this problem. It considers the similarity with all clusters of the tree branch at once and ranks the set of tree branches instead of single clusters on each level. Tree branches and lowest-level clusters have one-to-one correspondences, so we further do not differentiate the rankings of branches and rankings of the lowest-level clusters. We also refer to the branch that the contains the lowest-level cluster $c_{h,k}$ as branch number k; see Fig. 3.

Let $\boldsymbol{\theta}_k \in \mathbb{R}^h$ be the weight vector for branch k. Element θ_k^ℓ of this vector denotes the importance of the level ℓ cluster in the branch for classification. In general, if branches $k_1, \ldots, k_n$ contain internal cluster $c_{\ell,k}$, then there is a set of weights $\{\theta_{k_1}^\ell, \ldots, \theta_{k_n}^\ell\}$ that corresponds to this cluster and these weights can be different.

Let $\boldsymbol{\mu}_{\ell,k}$ be the mean vector of the parent cluster $B^{h-\ell}(c_{h,k})$ of cluster $c_{h,k}$

$$\boldsymbol{\mu}_{\ell,k} = \boldsymbol{\mu}\big(B^{h-\ell}(c_{h,k})\big).$$

Placing all these vectors for branch k together gives us the mean vector matrix $\mathbf{M}_k$. Column ℓ of this matrix corresponds to the parent cluster $B^{h-\ell}(c_{h,k})$ and equals $\boldsymbol{\mu}_{\ell,k}$:

$$\mathbf{M}_k = [\boldsymbol{\mu}_{1,k}, \ldots, \boldsymbol{\mu}_{h,k}].$$

We define the weighted hierarchical similarity $s_{\mathsf{h}}(\mathbf{x},\, c_{h,\,k})$ of a document $\mathbf{x}$ and the lowest level cluster $c_{h,\,k}$ as the weighted sum of similarities of the document $\mathbf{x}$ and clusters $c_{\ell,\,k}$ of the branch k

$$s_{\mathsf{h}}(\mathbf{x},\, c_{h,\,k}) = \sum_{\ell=1}^{h} \theta_k^\ell s\big(\mathbf{x},\, B^{h-\ell}(c_{h,\,k})\big) \equiv \sum_{\ell=1}^{h} \theta_k^\ell \mathbf{x}^\intercal \boldsymbol{\Lambda} \boldsymbol{\mu}_{\ell,k} \equiv \mathbf{x}^\intercal \boldsymbol{\Lambda} \mathbf{M}_k \boldsymbol{\theta}_k. \tag{14}$$

The document should be similar to all clusters of the branch to be similar with the lowest level cluster of this branch.

3 Model and parameter estimation

The hierarchical similarity function contains two sets of parameters: the parameter vector $\boldsymbol{\alpha}$ of the entropy model and the set of branches weight vectors $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}$. In this section, we describe a way to optimize these parameters, directly maximizing the quality AUCH (5) of the relevance operator.

Let $D_{\mathcal{V}_0} \cup D_{\mathcal{V}_1} \cup D_{\mathcal{V}_2}$ be a disjoint subsets of a training set D. We set the initial values of the parameters as

$$\boldsymbol{\alpha} = \mathbf{0}, \quad \boldsymbol{\theta}_k = \left[\frac{1}{h}, \ldots, \frac{1}{h}\right]. \tag{15}$$

The optimization algorithm alternates between the following steps:

1) find optimal values of $\boldsymbol{\alpha}$ given fixed values of $\boldsymbol{\theta}_k$ using subset $D_{\mathcal{V}_1}$,
2) find optimal values of $\boldsymbol{\theta}_k$ given fixed values of $\boldsymbol{\alpha}$ using subset $D_{\mathcal{V}_2}$.

In the next subsections we describe each of these steps in more detail.

Optimization of entropy model parameters $\boldsymbol{\alpha}$. We calculate the mean vectors $\{\boldsymbol{\mu}(c_{\ell,k})\}$ of the clusters using subset $D_{\mathcal{V}_0}$. These vectors provide estimates of the word entropy using (10) and (11) for each level of the hierarchy. We find the optimal $\alpha_1, \ldots, \alpha_h$ parameters of the entropy model (13) by solving the AUCH(R) (5) maximization task using the training subset $D_{\mathcal{V}_1}$:

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \text{AUCH}(R). \tag{16}$$

For a small number of levels h, this can be done via a grid search. Changing $\boldsymbol{\alpha}$ leads to a new $\boldsymbol{\Lambda}$ value, so after each iteration, we should renormalize the document vectors $\mathbf{x}$ to preserve the $\mathbf{x}^\intercal\boldsymbol{\Lambda}\mathbf{x} = 1$ condition and recalculate the mean vectors $\boldsymbol{\mu}(c_{\ell,k})$.

Optimization of the weight vectors $\{\boldsymbol{\theta}_k\}$. Given the training subset $D_{\mathcal{V}_2}$, we need to find the set of $\{\boldsymbol{\theta}_k\}$ that maximizes the hierarchical similarity of documents from $D_{\mathcal{V}_2}$ with their expert clusters. We keep $\boldsymbol{\alpha}$ fixed, so the values of $\boldsymbol{\Lambda}$ and $\mathbf{x}^\intercal\boldsymbol{\Lambda}\mathbf{M}_k$ are known for all documents $\mathbf{x}$. This leads to a convex quadratic programming task that is solved using the interior point method:

$$\boldsymbol{\theta}_k^* = \arg\max_{\boldsymbol{\theta}_k} \sum_{\mathbf{x}\in c_{h,k}} \mathbf{x}^\intercal\boldsymbol{\Lambda}\mathbf{M}_k\boldsymbol{\theta}_k + \psi\|\boldsymbol{\theta}_k - \mathbf{h}\|_2^2, \tag{17}$$

$$\|\boldsymbol{\theta}_k\|_1 = 1, \quad \boldsymbol{\theta}_k \geq \mathbf{0}, \quad k \in \{1 \ldots K_h\}, \quad \mathbf{h} = \left[\frac{1}{h}, \ldots, \frac{1}{h}\right]^\intercal, \tag{18}$$

where ψ is the regularization parameter. We should keep $\psi \neq 0$; otherwise, we face overfitting because (17) becomes a linear programming task and optimal solution will be a vertex of the simplex which makes one element of each $\boldsymbol{\theta}_k$ equal 1 and all others equal 0.

The complexity of this algorithm is

$$O(ba^h|D||W|hK_h), \tag{19}$$

where b is the number of steps 2 and 3 and a is the number of α_ℓ different values in the optimization grid. Experiments showed convergence in $b \sim 10$ steps. Code in [2] shows an example of estimating the parameters of hierarchical similarity function.

4 Bayesian approach in parameter estimation

The quality criterion AUCH (5) is based on the ranking and has discrete values. It restricts the set of possible optimization approaches. Still, the valid one from section 3 has an exponential increase in complexity (19) with the number of hierarchy levels h. It also involves dividing the training set into three subsets and decreases the number of objects available for optimization with respect to each set of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$. In this section, we use likelihood instead of AUCH to use more effective optimization methods. We define the likelihood of the document class matrix $\mathbf{Z}$ as

$$L(\mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\alpha}) = \prod_{n=1}^{N} \prod_{k=1}^{K_h} p^{z_{nk}}(z_{nk} = 1|\mathbf{x}_n, \boldsymbol{\theta}_k, \boldsymbol{\alpha}), \tag{20}$$

where the probability of class $c_{h,k}$ given document $\mathbf{x}_n$ is calculated using a softmax function of the weighted hierarchical similarity values $s_{n,k}$ of document $\mathbf{x}$ and tree branch k:

$$p(z_{nk} = 1|\mathbf{x}_n, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \frac{\exp(s_{n,k})}{\sum_{k'=1}^{k_h} \exp(s_{n,k'})}, \quad s_{n,k} = s_{\mathsf{h}}(\mathbf{x}_n, c_{h,k}|\boldsymbol{\theta}_k, \boldsymbol{\alpha}). \tag{21}$$

We assume that the parameters $\{\boldsymbol{\theta}_k\}$ and $\boldsymbol{\alpha}$ are random variables with the following distributions

$$p(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha}|\mathbf{0}, a^{-1}\mathbf{I}), \quad p(\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{\theta}_k|\mathbf{m}_k, \mathbf{V}_k^{-1}). \tag{22}$$

These priors normalize the values of $\boldsymbol{\alpha}$ and $\{\boldsymbol{\theta}_k\}$ and take into account our assumptions regarding them. Vector $\boldsymbol{\alpha}$ determines the influence of the word entropy on the word importance. A zero value of $\boldsymbol{\alpha}$ leads to the equal importance of all words. The weights vector $\boldsymbol{\theta}_k$ determines the weights of clusters in the branch k and has unknown expectation and covariance matrix. We put another prior on these hyperparameters

$$p(\mathbf{m}_k|\mathbf{V}_k) = \mathcal{N}(\mathbf{m}_k|\mathbf{m}_0, (b\mathbf{V}_k)^{-1}), \quad p(\mathbf{V}_k) = \mathcal{W}(\mathbf{V}_k|\mathbf{W}, \nu), \tag{23}$$

where $\mathcal{W}$ is a Wishart distribution. The mean vector $\mathbf{m}_0$ sets the initial assumption that clusters of each branch have the same weight $\mathbf{m}_{0,k} = 1/h$. The same idea was used for regularization in the previous section (15). The difference from regularization (18) is that elements of $\boldsymbol{\theta}_k$ now do not have to sum to 1. Still, we want to preserve the assumption that an increase in the weight of one cluster in the branch leads to a decrease in the others' weights. The Wishart parameter matrix $\mathbf{W}$ determines the covariance matrices $(b\mathbf{V}_k)^{-1}$ of $\boldsymbol{\theta}_k$. We define the $\mathbf{W}$ initial value in that manner to obtain negative correlations between $\boldsymbol{\theta}_k$ elements. We propose a joint model of document classes $\mathbf{Z}$, parameters $\boldsymbol{\theta}, \boldsymbol{\alpha}$, and hyperparameters $\mathbf{m}, \mathbf{V}$ as

$$p(\mathbf{Z}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) = L(\mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\alpha})p(\boldsymbol{\theta}|\mathbf{m}, \mathbf{V})p(\mathbf{m}|\mathbf{V})p(\mathbf{V})p(\boldsymbol{\alpha}). \tag{24}$$

Estimation of cluster probability given a document. Let $\tilde{\mathbf{Z}}$ be the class matrix for unlabelled documents (2). The relevance operator R ranks clusters of the lowest level according to the probability of the cluster given the document. We use model (24) to find the posterior distribution of the hierarchical similarity parameters $\boldsymbol{\alpha}$ and $\{\boldsymbol{\theta}_k\}$ and estimate these probabilities. Two possible types of estimates are found as follows: 1) use the maximum posterior values of the parameters $\boldsymbol{\theta}_k^{\text{MAP}}, \boldsymbol{\alpha}^{\text{MAP}}$ and calculate the probability as a softmax value (25) of similarities, or 2) calculate the evidence estimate (26). In this paper we use the second approach because it takes into account the shape of the posterior distribution and yields better estimates.

$$p(\tilde{z}_{tk} = 1|\tilde{\mathbf{x}}_t) = p(\tilde{z}_{tk}|\tilde{\mathbf{x}}_t, \boldsymbol{\theta}_k^{\text{MAP}}, \boldsymbol{\alpha}^{\text{MAP}}) \tag{25}$$

$$p(\tilde{z}_{tk} = 1|\tilde{\mathbf{x}}_t) = \int p(\tilde{z}_{tk}|\tilde{\mathbf{x}}_t, \boldsymbol{\theta}, \boldsymbol{\alpha})p(\boldsymbol{\theta}, \boldsymbol{\alpha}|\mathbf{Z})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{\alpha} \tag{26}$$

It is not possible to calculate the posterior distribution of parameters due to the non linearity of the likelihood L (20) on $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$. We use variational inference to obtain the posterior estimate [6, 9, 14] q. Integration (26) also does not have a closed form solution due to the softmax function structure. To avoid multiple similar approximations, we at once approximate the joint posterior distribution $p(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}|\mathbf{Z})$ of the unlabelled document classes $\tilde{\mathbf{Z}}$, parameters, and hyperparameters instead of the regular posterior $p(\boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}|\mathbf{Z})$. The joint posterior distribution is the entire expression under the integral (26) and it's approximation allows us to calculate the integral analytically.

The joint distribution of the proposed model (24) and unlabelled document classes $\tilde{\mathbf{Z}}$ is defined by the following equation

$$p(\tilde{\mathbf{Z}}, \mathbf{Z}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) = p(\tilde{\mathbf{Z}}|\boldsymbol{\theta}, \boldsymbol{\alpha})p(\mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{m}, \mathbf{V}). \tag{27}$$

Let $q(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha})$ be an approximation of the joint posterior $p(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}|\mathbf{Z})$. We use mean field approximation (29) and search for the optimal q that minimizes the KL divergence:

$$\mathrm{KL}(q(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha})\|p(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}|\mathbf{Z})) \to \min_{q = q(\boldsymbol{\theta})q(\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V})q(\tilde{\mathbf{Z}})}. \tag{28}$$

$$q(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) = q(\boldsymbol{\theta})q(\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V})q(\tilde{\mathbf{Z}}). \tag{29}$$

As stated in [5], KL minimization (28) is equivalent to maximization of the lower bound $\mathcal{L}(q)$:

$$\mathcal{L}(q) = \int q(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) \ln \left(\frac{p(\tilde{\mathbf{Z}}, \mathbf{Z}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha})}{q(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha})} \right) \mathrm{d}\boldsymbol{\theta} \mathrm{d}\mathbf{m} \mathrm{d}\mathbf{V} \mathrm{d}\boldsymbol{\alpha} \mathrm{d}\tilde{\mathbf{Z}} \to \max_{q = q(\boldsymbol{\theta}) q(\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V}) q(\tilde{\mathbf{Z}})}. \quad (30)$$

To find the optimal factors of q, we solve (30) according to one factor of q, keeping all other factors constant. This leads to the following form of factors

$$\begin{aligned} \ln q(\boldsymbol{\theta}) &= \mathsf{E}_{\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V}, \tilde{\mathbf{Z}}} \big[\ln p(\tilde{\mathbf{Z}}, \mathbf{Z}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) \big] + \mathrm{const}(\boldsymbol{\theta}), \\ \ln q(\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V}) &= \mathsf{E}_{\boldsymbol{\theta}, \tilde{\mathbf{Z}}} \big[\ln p(\tilde{\mathbf{Z}}, \mathbf{Z}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) \big] + \mathrm{const}(\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V}), \\ \ln q(\tilde{\mathbf{Z}}) &= \mathsf{E}_{\boldsymbol{\alpha}, \mathbf{m}, \mathbf{V}, \boldsymbol{\theta}} \big[\ln p(\tilde{\mathbf{Z}}, \mathbf{Z}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) \big] + \mathrm{const}(\tilde{\mathbf{Z}}), \end{aligned} \quad (31)$$

where $\mathrm{const}(var)$ defines some expression that does not depend on var. Iterative recalculation of these factors leads to the maximum because each iteration does not decrease the $\mathcal{L}(q)$ value [5].

The likelihood L (20) contains the sum of the exponents of the random variables $\boldsymbol{\theta}_k$ and $\boldsymbol{\alpha}$, so we cannot calculate the factor estimates (31) analytically. We use the method of local variations [10] to approximate the likelihood with it's upper bound. Let $g(\mathbf{x})$ be the sum of expectations

$$g(\mathbf{x}) = \sum_{k=1}^{K_h} \exp(x_k). \quad (32)$$

The expression $-\ln\big(g(\mathbf{x})\big)$ is a convex function (see Fig. 4), so the tangent plane through some point $\boldsymbol{\xi}$ is an upper bound for this expression

$$y(\mathbf{x}, \boldsymbol{\xi}) = -\ln\big(g(\boldsymbol{\xi})\big) - \nabla \ln\big(g(\boldsymbol{\xi})\big)^{\mathsf{T}} (\mathbf{x} - \boldsymbol{\xi}), \quad -\ln\big(g(\boldsymbol{\xi})\big) \le y(\mathbf{x}, \boldsymbol{\xi}). \quad (33)$$

Taking the exponent from both sides of inequality (33), we obtain the upper bound of one over $g(\mathbf{x})$

$$\frac{1}{g(\mathbf{x})} \le \frac{1}{g(\boldsymbol{\xi})} \exp\left(\sum_{k=1}^{K_h} \frac{\exp(\xi_k)}{g(\boldsymbol{\xi})} (\xi_k - x_k) \right). \quad (34)$$

The index of the exponent on the right side of (34) is a linear function of $\mathbf{x}$. The product of this bound and the density functions from the exponential family leaves it inside the exponential class and makes calculation of the expectation straightforward.

We obtain an upper bound of $\mathcal{L}(q)$ using the constructed approximation (34) of the softmax denominator for each document $\mathbf{x}_n$:

$$\mathcal{L}(q) \le \hat{\mathcal{L}}(q, \boldsymbol{\xi}), \quad (35)$$

where $\boldsymbol{\xi} = \{\boldsymbol{\xi}_n\}$ is the set of variational parameters. We minimize $\hat{\mathcal{L}}(q, \boldsymbol{\xi})$ according to $\boldsymbol{\xi}$ to find the closest upper bound of $\mathcal{L}(q)$.

The optimal factors (31) calculated for the joint model (27) with the softmax approximation (34) have the following form

$$\begin{aligned} q(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}) &= q(\boldsymbol{\alpha}) \prod_{k=1}^{K_h} q(\boldsymbol{\theta}_k) q(\mathbf{m}_k | \mathbf{V}_k) q(\mathbf{V}_k) \prod_{t=1}^{|T|} q(\tilde{z}_{tk}), \\ q(\boldsymbol{\alpha}) &\sim \mathcal{N}(\boldsymbol{\alpha}_0, a^{-1}\mathbf{I}), \\ q(\boldsymbol{\theta}_k) &\sim \mathcal{N}(\mathbf{m}'_{0k}, (\nu' \mathbf{V}_k)^{-1}), \\ q(\mathbf{m}_k | \mathbf{V}_k) q(\mathbf{V}_k) &\sim \mathcal{N}(\mathbf{m}_{0k}, (b' \mathbf{V}_k)^{-1}) \mathcal{W}(\mathbf{W}_k, \nu'), \\ q(\tilde{z}_{tk}) &\sim \mathrm{Bern}(p_{tk}). \end{aligned} \quad (36)$$

Parameters ν' and b' equal $\nu' = \nu + 1, \quad b' = 1 + b$, and parameters $\mathbf{m}_{0k}, \mathbf{W}_k, \boldsymbol{\alpha}_0, \mathbf{m}_{0k'}$, and p_{tk} are recalculated iteratively using

$$\begin{aligned}
\mathbf{m}_{0k} &= \frac{\mathsf{E}\boldsymbol{\theta}_k + b\mathbf{m}_0}{b'}, \quad \mathbf{W}_k^{-1} = b'\mathbf{m}_{0k}\mathbf{m}_{0k}^\intercal + b\mathbf{m}_0\mathbf{m}_0^\intercal + \mathsf{E}\left[\boldsymbol{\theta}_k\boldsymbol{\theta}_k^\intercal\right] + \mathbf{W}^{-1}, \\
\boldsymbol{\alpha}_0 &= \frac{1}{a}\sum_{m=1}^{|W|}\boldsymbol{\iota}_m\sum_{k=1}^{K_h}(\mathbf{M}_k\mathsf{E}\boldsymbol{\theta}_k)_m\left(\sum_{n=1}^{N}x_{nm}\hat{z}_{nk} + \sum_{t=1}^{T}\tilde{x}_{tm}\hat{\tilde{z}}_{tk}\right), \\
\mathbf{m}'_{0k} &= \mathbf{m}_{0k} + \frac{1}{\nu'}\mathbf{W}_k^{-1}\mathbf{M}_k^\intercal\mathsf{E}_{\boldsymbol{\alpha}}\boldsymbol{\Lambda}\left(\sum_{n=1}^{N}\mathbf{x}_n\hat{z}_{nk} + \sum_{t=1}^{T}\tilde{\mathbf{x}}_t\hat{\tilde{z}}_{tk}\right), \\
p_{tk} &= \frac{\exp(\zeta_{tk})}{\exp(\zeta_{tk}) + g(\tilde{\boldsymbol{\xi}}_t)},
\end{aligned} \tag{37}$$

where we defined $\hat{z}_{nk}, \hat{\tilde{z}}_{tk}$ and ζ_{tk} to make the formulas uncluttered:

$$\begin{aligned}
\hat{z}_{nk} &= \left(z_{nk} - \frac{\exp(\xi_{nk})}{g(\boldsymbol{\xi}_n)}\right), \quad \hat{\tilde{z}}_{tk} = \left[\mathsf{E}\tilde{z}_{tk} - \frac{\exp(\tilde{\xi}_{tk})}{g(\tilde{\boldsymbol{\xi}}_t)}\left(\sum_{k'=1}^{K_h}\mathsf{E}\tilde{z}_{tk'}\right)\right], \\
\zeta_{tk} &= \tilde{\mathbf{x}}_t^\intercal\mathsf{E}_{\boldsymbol{\alpha}}\boldsymbol{\Lambda}\mathbf{M}_k\mathsf{E}\boldsymbol{\theta}_k + \sum_{k'=1}^{K_h}\frac{\exp(\tilde{\xi}_{tk'})}{g(\tilde{\boldsymbol{\xi}}_t)}(\tilde{\xi}_{tk'} - \tilde{\mathbf{x}}_t^\intercal\mathsf{E}_{\boldsymbol{\alpha}}\boldsymbol{\Lambda}\mathbf{M}_{k'}\mathsf{E}\boldsymbol{\theta}_{k'}).
\end{aligned} \tag{38}$$

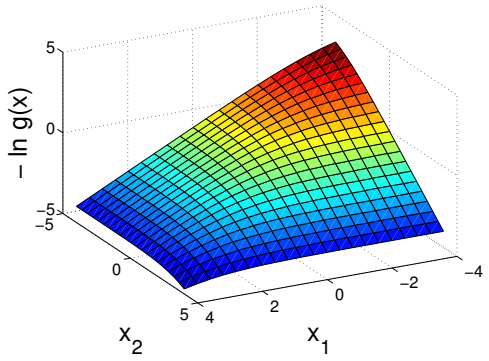


Fig. 4: Values of $\tilde{g} = -\ln g(\mathbf{x})$ for the two-dimensional $\mathbf{x}$.

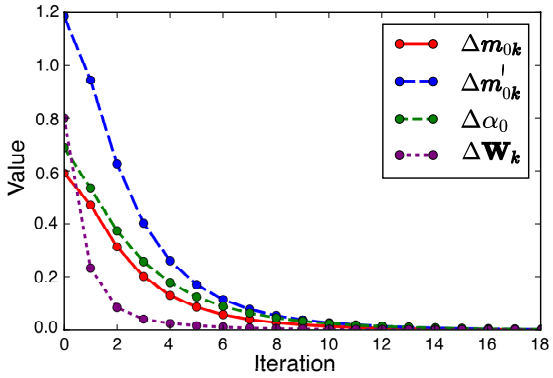


Fig. 5: Example of parameter convergence.

EM algorithm for parameter optimization. To find the best approximation of the joint posterior $p(\tilde{\mathbf{Z}}, \boldsymbol{\theta}, \mathbf{m}, \mathbf{V}, \boldsymbol{\alpha}|\mathbf{Z})$ we iteratively recalculate each factor of q keeping the other fixed according to (31).

The likelihood upper bound (34) with variational parameters $\boldsymbol{\xi} = \{\boldsymbol{\xi}_n\}$ and $\tilde{\boldsymbol{\xi}} = \{\tilde{\boldsymbol{\xi}}_t\}$ allows us to calculate the closed-form solution (37) for each factor using parameters of other factors. This leads to the EM algorithm, which alternates step E to calculate the q parameters using (37) with step M to optimize the variational parameters $\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}$ of the $\mathcal{L}(q)$ upper bound.

1. Initialize the parameters

$$\mathbf{W}, \nu, \mathbf{m}_0, a, b, \mathbf{W}_k = \mathbf{W}, \nu' = \nu + 1, b' = b + 1, \mathbf{m}_{0k} = \mathbf{m}_0, \boldsymbol{\xi}_n.$$

2. Calculate $\mathsf{E}\boldsymbol{\theta}_k, \mathsf{E}[\boldsymbol{\theta}_k\boldsymbol{\theta}_k^\intercal]$ according to the $q(\boldsymbol{\theta}_k)$ distributions

$$\begin{aligned}
\mathsf{E}\boldsymbol{\theta}_k &= \mathbf{m}'_{0k}, \\
\mathsf{E}[\boldsymbol{\theta}_k\boldsymbol{\theta}_k^\intercal] &= (\nu'\mathbf{W}_k)^{-1} + \mathbf{m}'_{0k}(\mathbf{m}'_{0k})^\intercal,
\end{aligned} \tag{39}$$

and recalculate the parameters of the $q(\mathbf{m}), q(\mathbf{V}), q(\boldsymbol{\alpha})$ factors using (37).

3. Calculate $\mathsf{E}_{\boldsymbol{\alpha}}\boldsymbol{\Lambda}$ using the $q(\boldsymbol{\alpha})$ distribution

$$\mathsf{E}_{\boldsymbol{\alpha}}\boldsymbol{\Lambda} = \tilde{\boldsymbol{\Lambda}} = \operatorname{diag}(\{\lambda'_m\}), \quad \lambda'_m = 1 + \boldsymbol{\alpha}_0^{\intercal}\boldsymbol{\iota}_m, \tag{40}$$

and recalculate parameters of $q(\boldsymbol{\theta}_k)$ factors using (37).

4. Optimize the variational parameters

$$\xi_{nk} = \mathbf{x}_n^{\intercal}\tilde{\boldsymbol{\Lambda}}\mathbf{M}_k\mathbf{m}'_{0k}, \quad \tilde{\xi}_{tk} = \tilde{\mathbf{x}}_t^{\intercal}\tilde{\boldsymbol{\Lambda}}\mathbf{M}_k\mathbf{m}'_{0k}. \tag{41}$$

If some of parameters have changed significantly in steps 2-4, go back to step 2.

Probability of a class given a document. The optimal joint posterior approximation has the form (36), where the distribution of the class $c_{h,k}$ label $\tilde{z}_{tk}$ for a document $\tilde{\mathbf{x}}_t$ is a Bernoulli distribution with parameter p_{tk} (38). The integral from the joint posterior (26) gives a Bayesian estimate of the cluster probability. Substitution of (36) into (26) gives a straightforward estimate of the probability $p(\tilde{z}_{tk} = 1|\tilde{\mathbf{x}}_t) = p_{tk}$. For each document, the relevance operator R ranks clusters according to this estimate.

5 Computational Experiment

To test the proposed approach and compare it with well-known methods we solve a hierarchical classification task for two text collections: abstracts of the EURO conference and web-sites of industry companies.

Collection of EURO abstracts. We used programs of the scientific conference EURO from 2006 through 2016 [1]. To unify data from conferences of different years and to build a single structure for collection we used the following procedure.

1. Take an expert cluster structure of EURO 2016 as a Base (Fig. 1).
2. For each cluster c of the EURO 2010-2015 conferences search for the same cluster in the Base structure. If the one is found, merge c with it; if there is no such cluster, add c as a new one to the Base structure.
3. For each cluster c of the EURO 2006-2009 conferences search for the same cluster in the Base structure. If one is found, merge c with it; if there is no such cluster, discard all documents from c.

The joint collection contains $|D| = 15527$ documents, the dictionary contains $|W| = 24304$ words, and the Base hierarchical structure consists of $K_2 = 26$ clusters of the second level (Area) and $K_3 = 264$ clusters of the third level (Stream).

Table 1: Ranking quality AUCH (5) of the different algorithms and training set sizes $|D_{\mathcal{V}}|$.

Algorithm \ $\lvert D_{\mathcal{V}}\rvert$	500	1000	1500	3000	5000	7000	10000
svm	0.76	0.80	0.81	0.84	0.85	0.86	0.87
hNB	0.77	0.82	0.84	0.87	0.90	0.91	0.92
suhiPLSA	0.75	0.79	0.80	0.81	0.82	0.84	0.84
hSim	0.80	**0.86**	**0.88**	**0.90**	**0.91**	**0.92**	**0.93**
hSimWV	**0.82**	**0.86**	0.87	**0.89**	**0.92**	**0.92**	0.92

Ranking results for unlabelled documents. For the ranking experiment, we considered Area and Stream levels of hierarchy. We constructed a relevance operator R using the proposed weighted hierarchical similarity function hSim (14) and used the EM algorithm from section 4 to optimize it's parameters on the training subset. The results of this function were compared with those of other algorithms for hierarchical ranking: 1) hierarchical naive Bayes hNB [22], 2) probabilistic regularized model SuhiPLSA [18] and 3) hierarchical multiclass svm [12].

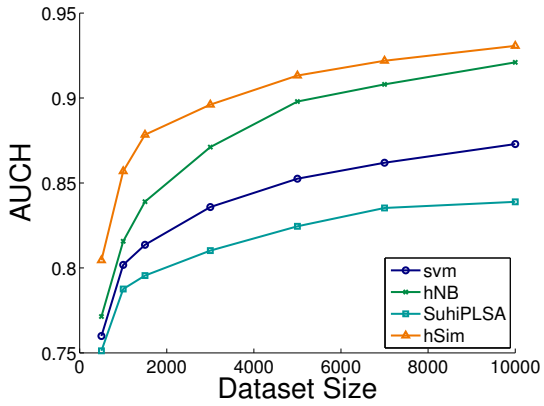
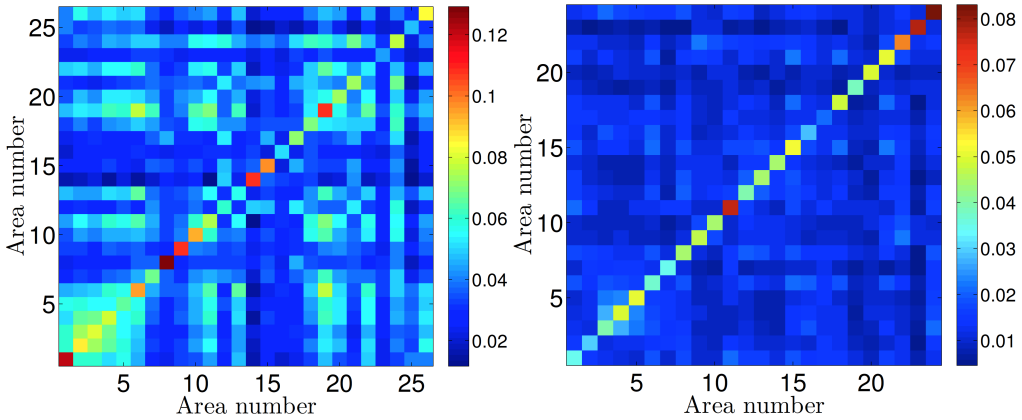


Fig. 6: Test AUCH quality dependence on the training sample size for the svm, hNB, suhiPLSA, and hSim algorithms.

We divided collection D into two parts: train $D_{\mathcal{V}}$ and test $D_{\mathcal{T}}$ in different proportions. The size of the training subset $|D_{\mathcal{V}}|$ varied from 500 documents to 10000. The size of the testing subset $D_{\mathcal{T}}$ was fixed, $|D_{\mathcal{T}}| = 5000$. Each algorithm was trained on $D_{\mathcal{V}}$ and returned a ranked list of clusters for a given document. The qualities of the algorithms were measured using the area under the cumulative histogram AUCH (5).



(a) Same importance for all words, $\mathbf{\Lambda} = \mathbf{I}$ (b) Optimized importance, $\mathbf{\Lambda} = \mathbf{\Lambda}^*$

Fig. 7: Pairwise similarities of Area level clusters with the entropy model (b) and without (a).

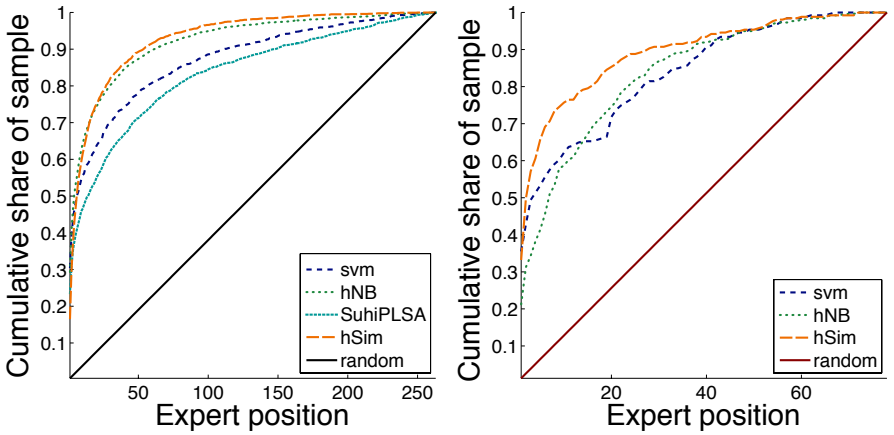
Fig. 5 shows the convergence of parameters during optimization with the EM algorithm from section 4. Table 1 contains the values of AUCH for all algorithms and the sizes of the training samples. Bold values correspond to the best statistically equivalent values for each training sample size. Fig. 6 shows the table data in chart format. The proposed hSim algorithm exhibited the best results. Fig. 8a. shows the envelope curve for the cumulative histogram (4) for the training sample size $|D_{\mathcal{V}}| = 10000$.

Fig. 7 demonstrates the effect of the entropy model. It visualizes the matrix of pair similarity of expert clusters on the Area level of the hierarchy. We suppose that expert clustering is an optimal solution, so the similarity function should separate intracluster similarity and intercluster similarity well. The right part of Fig. 7 shows the values of weighted cluster similarity that uses $\mathbf{\Lambda} = \mathbf{\Lambda}^*$

Table 2: AUCH (5) values for different algorithms. Collection of industry companies web sites.

Algorithm	AUCH
svm	0.83
hNB	0.83
hSim	**0.89**

with the optimal entropy model parameter $\boldsymbol{\alpha}$, and the left part of Fig. 7 shows the values of clusters similarity without optimization, $\boldsymbol{\Lambda} = \mathbf{I}$. We can see from the figures that intracluster similarities (diagonal elements of the matrix) became greater than intercluster similarities (non-diagonal elements) after optimization. The optimal $\boldsymbol{\Lambda}^*$ corresponds to a 0.047 average intracluster similarity and 0.012 average intercluster similarity.



(a) Collection of EURO conference abstracts, 10000 training objects

(b) Collection of industry companies web sites objects

Fig. 8: Envelope curves of cumulative histograms for different algorithms.

Collection of industrial companies web sites. In this collection each web site is represented by a set of HTML pages. We merge all pages into one and remove all special symbols and tags to form a single text document for each site. The final collection contains $|D| = 1036$ documents, the dictionary contains $|W| = 18775$ words, and the hierarchical structure contains $K_2 = 11$ clusters of the second level and $K_3 = 78$ clusters of the third level.

The training subset $D_{\mathcal{V}}$ consists of 750 documents, and the test subset $D_{\mathcal{T}}$ consists of the remaining 286 documents. We compare the results of the proposed weighted hierarchical similarity function hSim with those of 1) hierarchical naive Bayes hNB [22] and 2) hierarchical multiclass svm [12]. Table 2 shows the AUCH (5) quality criterion values for these algorithms. Fig. 8b. shows the corresponding envelope of the cumulative histograms (4).

6 Conclusion

In this paper, we solve a hierarchical text classification task for partly labelled collections with a tree cluster structure given by experts. To find the relevance of the clusters to the given document, we propose a weighted hierarchical similarity function of a document and a branch of the cluster structure. This function allows the ranking of entire branches of the hierarchy instead of using the common top-down approach. The proposed function contains two sets of parameters: word

importance for classification and weight vectors for each branch of the cluster tree. To estimate the importance of the words, we propose a model that calculates a word's importance using its entropy.

To use effective optimization techniques, we propose a joint probabilistic model of document classes, parameters and hyperparameters. Variational Bayesian inference and the likelihood upper bound allow us to approximate the joint posterior distribution of unlabelled document classes and parameters and calculate the Bayesian estimate of a class probability given a document. The proposed relevance operator ranks clusters according to the probability estimates. We compare the results of our approach with those of hierarchical multiclass SVM, hierarchical naive Bayes and Suhi PLSA on two types of text collections: abstracts of the major conference EURO and web sites of industry companies. The proposed approach exhibited comparable results on both collections.

For future work, we are going to use other types of likelihood bounds, such as quadratic lower bounds; compare the variational inference results with sampling techniques; and generalize hierarchical similarity to other types of cluster structures, such as directed acyclic graphs.

References

1. Collection of the euro and ifors abstracts, https://sourceforge.net/p/mlalgorithms/code/HEAD/tree/PhDThesis/Kuzmin/Data/EURO/. (last checked: 31.03.2019)
2. Hierarchical similarity model matlab code., https://svn.code.sf.net/p/mlalgorithms/code/KuzminCICLing2019/code/. (last checked: 31.03.2019)
3. Alexandrov, M., Gelbukh, A., Rosso, P.: An approach to clustering abstracts. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) Natural Language Processing and Information Systems. pp. 275–285. Springer Berlin Heidelberg (2005)
4. Cordeiro de Amorim, R., Mirkin, B.: Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. Pattern Recogn. **45**(3), 1061–1075 (2012)
5. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg (2006)
6. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: Review for statisticians. CoRR **abs/1601.00670** (2016)
7. Frank, E., Bouckaert, R.R.: Naive bayes for text classification with unbalanced classes. In: Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases. pp. 503–510. PKDD'06, Springer-Verlag, Berlin, Heidelberg (2006)
8. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale bayesian logistic regression for text categorization. Technometrics **49**, 291–304 (2007)
9. Gershman, S., Hoffman, M.D., Blei, D.M.: Nonparametric variational inference. In: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK (2012)
10. Gibbs, M.: Bayesian Gaussian Processes for Regression and Classification. Ph.D. thesis, Cambridge University (1997)
11. Gong, L., Zeng, J., Zhang, S.: Text stream clustering algorithm based on adaptive feature selection. Expert Systems with Applications **38**(3), 1393–1399 (2011)
12. Hao, P.Y., Chiang, J.H., Tu, Y.K.: Hierarchically svm classification based on support vector clustering method and its application to document categorization. Expert Systems with Applications **33**(3), 627–635 (2007)
13. He, Q., Chang, K., Lim, E.P., Banerjee, A.: Keep it simple with time: A reexamination of probabilistic topic detection models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(10), 1795–1808 (2010)
14. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. J. Mach. Learn. Res. **14**(1), 1303–1347 (2013)
15. Ikonomakis, M., Kotsiantis, S., Tampakas, V.: Text classification using machine learning techniques. WSEAS Transactions on Computers **4**(8), 966–974 (2005)
16. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning. pp. 143–151 (1997)
17. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) Machine Learning: ECML-98. pp. 137–142. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)

18. Kuznetsov, M., Clausel, M., Amini, M., Gaussier, E., Strijov, V.: Supervised topic classification for modeling a hierarchical conference structure. In: Neural Information Processing 22nd International Conference, ICONIP 2015, Istanbul, Turkey, 2015, Proceedings, Part I. pp. 90–97 (2015)
19. Largeron, C., Moulin, C., Géry, M.: Entropy based feature selection for text categorization. In: Proceedings of the 2011 ACM Symposium on Applied Computing. pp. 924–928. SAC'11, ACM, New York, NY, USA (2011)
20. Leisch, F.: A toolbox for k-centroids cluster analysis. Comput. Stat. Data Anal. **51**(2), 526–544 (2006)
21. Li, C.H., Park, S.C.: Text categorization based on artificial neural networks. In: King, I., Wang, J., Chan, L.W., Wang, D. (eds.) Neural Information Processing. pp. 302–311. Springer Berlin Heidelberg (2006)
22. McCallum, A., Rosenfeld, R., Mitchell, T.M., Ng, A.Y.: Improving text classification by shrinkage in a hierarchy of classes. In: Proceedings of the Fifteenth International Conference on Machine Learning. pp. 359–367. ICML '98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)
23. Mikawa, K., Ishida, T., Goto, M.: A proposal of extended cosine measure for distance metric learning in text classification. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics. pp. 1741–1746 (2011)
24. Ruiz, M.E., Srinivasan, P.: Hierarchical text categorization using neural networks. Information Retrieval **5**(1), 87–118 (2002)
25. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)
26. Schedl, M.: Nowplaying madonna: A large-scale evaluation on estimating similarities between music artists and between movies from microblogs. Inf. Retr. **15**(3-4), 183–217 (2012)
27. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. **6**, 1453–1484 (2005)
28. Xue, G.R., Xing, D., Yang, Q., Yu, Y.: Deep classification in large-scale text hierarchies. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 619–626 (2008)
29. Yih, W.t.: Learning term-weighting functions for similarity measures. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2. pp. 793–802. EMNLP '09 (2009)