Hierarchical Latent Tree Analysis for Topic Detection

Tengfei Liu, Nevin L. Zhang, and Peixian Chen

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{liutf,lzhang,pchenac}@cse.ust.hk

Abstract. In the LDA approach to topic detection, a topic is determined by identifying the words that are used with high frequency when writing about the topic. However, high frequency words in one topic may be also used with high frequency in other topics. Thus they may not be the best words to characterize the topic. In this paper, we propose a new method for topic detection, where a topic is determined by identifying words that appear with high frequency in the topic and low frequency in other topics. We model patterns of word co-occurrence and co-occurrences of those patterns using a hierarchy of discrete latent variables. The states of the latent variables represent clusters of documents and they are interpreted as topics. The words that best distinguish a cluster from other clusters are selected to characterize the topic. Empirical results show that the new method yields topics with clearer thematic characterizations than the alternative approaches.

1 Introduction

Topic models have been the focus of much research in the past decade. The predominant methods are latent Dirichlet allocation (LDA) [3] and its variants [2,11]. These methods assume a generating process for the documents. For example, to generate one document, LDA first samples a multinomial distribution over topics, then it repeatedly samples a topic according to this distribution and samples a word from the topic. In this setting, a topic is defined as a multinomial distribution over the entire vocabulary. Each document is viewed as a probabilistic mixture of all the topics. The topics and the topic composition of each document are inferred by inverting the generating process using statistical techniques such as variational inference [3] and Gibbs sampling [2].

The topic definition in LDA or its variants models how frequent an author would use each word in the vocabulary when writing about a topic. A few words with high frequency are usually selected to interpret the topic [3]. However, this does not consider the differences in word usage between the documents about the topic and the documents not about the topic. High frequency words in one topic might also appear with high frequency in other topics. They may be common words for multiple topics and contain little content information to the specific topic. Thus the high frequency words are not necessarily the best words to describe the topic. To better characterize a topic, it would be advisable to consider the words that appear with high probability in the documents about the topic, while appear with low probability in the documents not on the topic. We call such words the characteristic words of the topic.

T. Calders et al. (Eds.): ECML PKDD 2014, Part II, LNCS 8725, pp. 256–272, 2014.

When writing an article on a topic, an author is likely to use the characteristic words along with other non-characteristic words. When describing a topic, however, it would be better to focus on the characteristic words. For example, if we try to describe the topic *military*, we may use only a few words such as *troop*, *army*, *soldier*, *weapon*, *gun*, *bomb*, *tank*, *missile* and so on. To write an article on military, on the other hand, we might use many other words. The characteristic words of the topic consist of only a small fraction of all the words in an article on the topic.

In this paper, we propose a new method for topic detection that determines topics by identifying their characteristic words. The key idea is to model patterns of word co-occurrence and co-occurrence of those patterns using a hierarchy of discrete latent variables. Each latent variable represents a soft partition of the documents based on some word co-occurrence patterns. The states of the latent variable correspond to document clusters in the partition. They are interpreted as topics. Each document may belong to multiple clusters in different partitions. In other words, a document might belong to two or more topics 100% simultaneously. For each topic, the words that best distinguish the topic from other topics are selected to describe the topic. These words are usually the words appear with high probability in the documents belonging to the topic, while appear with low probability in the documents belonging to other topics.

This paper builds upon previous work on latent tree models (LTMs) by Zhang [19], which are tree-structured probabilistic graphical models where leaf nodes represent observed variables and internal nodes represent latent variables. When applied to text data, LTMs are effective in systematically discovering patterns of word co-occurrence [13,12]. In this work, we introduce semantically higher level latent variables to model co-occurrence of those patterns, resulting in hierarchical latent tree models (HLTMs). The latent variables at higher levels of the hierarchy correspond to more general topics, while the latent variables at lower levels correspond to more specific topics. The proposed method for topic detection is therefore called *hierarchical latent tree analysis (HLTA)*.

The remainder of this paper is organized as follows. In Section 2 we briefly introduce the concepts of LTMs. In Section 3 we present the HLTA algorithm. We will explain how to find the patterns of word co-occurrence and then how to aggregate these patterns for better topic detection. In Section 4, we present empirical results and compare HLTA with alternative methods. Finally, conclusions are drawn in Section 6.

2 Basics of Latent Tree Models

A *latent tree model (LTM)* is a Markov random field over an undirected tree where leaf nodes represent observed variables and internal nodes represent latent variables. LTMs were originally called hierarchical latent class models [19] to underline the fact that they are a generalization of *latent class model (LCM)* [1]. Figure 1 shows an example LTM that was learned from a collection of documents. The words at the bottom are binary variables that indicate the presence or absence of the words in a document. The Z_i's are the latent variables. They are discrete and their *cardinalities*, i.e., the number of states, are given in parentheses.

For technical convenience, we often root an LTM at one of its latent nodes and regard it as a directed graphical model, i.e., a *Bayesian network* [16]. Then all the

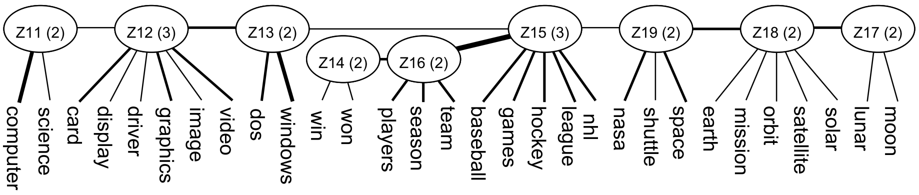

Fig. 1. A latent tree model for a toy text data set. The edge widths visually show the strength of correlation between variables. They are computed from the probability distributions of the model.

edges are directed away from the root. The numerical information of the model includes a marginal distribution for the root and one conditional distribution for each edge. For example, for edge $Z15 \rightarrow hockey$, it is associated with probability $P(hockey \mid Z15)$. The conditional distribution associated with each edge characterizes the probabilistic dependence between the two nodes that the edge connects. The product of all those distributions defines a joint distribution over all the latent and observed variables.

In general, suppose there are n observed variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ and m latent variables $\mathbf{Z} = \{Z_1, \ldots, Z_m\}$. Denote the parent of a variable Y as $pa(Y)$ and let $pa(Y)$ be a empty set when Y is the root. Then the LTM defines a joint distribution over all observed and latent variables as follows:

$$P(X_1, \ldots, X_n, Z_1, \ldots, Z_m) = \prod_{Y \in \mathbf{X} \cup \mathbf{Z}} P(Y \mid pa(Y)) \tag{1}$$

3 Topic Detection with Hierarchical Latent Tree Models

In this section, we describe the new method HLTA. Conceptually, the method has three steps: (1) Discover patterns of word co-occurrences; (2) Build a hierarchy by recursively discovering co-occurrence patterns of patterns; and (3) Extract topics from the resulting hierarchy. The pseudo code of the algorithm is given in Algorithm 1. We describe the steps in details in the following subsections.

3.1 Discovering Co-occurrence of Words

In HLTA, words are regarded as binary variables which indicate the presence or absence of the words in the documents. Documents are represented as binary vectors. The first step of HLTA is to identify the patterns of word co-occurrence. This is done by partitioning the word variables into clusters such that variables in each cluster are closely correlated and the correlations among variables in each cluster can be properly modeled with only one latent variable. The Bridged-Islands algorithm (BI) [12] is used for this purpose (Line 3).

Figure 1 shows the latent tree model that BI obtained on a toy data set. We see that the word variables are partitioned into 9 clusters. One example cluster is {*baseball*, *games*, *hockey*, *league*, *nhl*}. The five word variables are connected to the same latent variable $Z15$ and are hence called *siblings*. The cluster is then called a sibling cluster. It

Algorithm 1. HLTA(D_{in}, δ, k)

Inputs: D_{in}: input dataset; δ: Bayes factor threshold, k: maximum level of latent variables.
Output: An HLTM and the topics.

```
$D \leftarrow D_{in}$, $Level \leftarrow 0$, $M_{whole} \leftarrow \emptyset$ .
while $Level < k$ or $D$ contains more than two variables do
    $M \leftarrow$ Bridged-Islands(D, $\delta$)
    if $M_{whole} = \emptyset$ then
        $M_{whole} \leftarrow M$
    else
        $M_{whole} \leftarrow$ MergeModel($M_{whole}$, M)
    end if
    $D' \leftarrow$ ProjectData($M_{whole}$, $D_{in}$)
    $D \leftarrow D'$, $Level \leftarrow Level + 1$
end while
Output topics in different levels and return $M_{whole}$.
```

is apparent that the words in each sibling cluster are semantically correlated and tend to co-occur. The correlations among the word variables in each sibling cluster are modeled by a latent variable. In fact, every latent variable in the model is connected to at least one word variable. Because of this, the model is called a *flat latent tree model*.

In the following, we briefly describe how BI works. The reader is referred to [12] for the details. In general, BI is a greedy algorithm that aims at finding the flat latent tree model with the highest Bayesian Information Criterion (BIC) score [17]. It proceeds in four steps: (1) partition the set of variables into sibling clusters; (2) introduce a latent variable for each sibling cluster; (3) connect the latent variables to form a tree; (4) refine the model based on global considerations.

To identify potential siblings, BI considers how closely correlated each pair of variables are in terms of mutual information. The mutual information (MI) $I(X;Y)$ [8] between the two variables X and Y is defined as follows:

$$I(X;Y) = \sum_{X,Y} P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)}, \tag{2}$$

where the summation is taken over all possible states of X and Y. The distributions $P(X,Y)$, $P(X)$ and $P(Y)$ are estimated from data.

To determine the first sibling cluster, BI maintains a working set $\boldsymbol{S}$ of variables that initially consists of the pair of variables with the highest MI. Other variables are added to the set one by one. At each step, BI chooses to add the variable X that maximizes the quantity $\max_{Z \in \boldsymbol{S}} I(X;Z)$. After each step of expansion, BI performs a Bayesian statistical test to determine whether correlations among the variables in $\boldsymbol{S}$ can be properly modeled using one single latent variable. The test is called *uni-dimensionality test* or simply *UD-test*. The expansion stops when the UD-test fails.

To perform the UD-test, BI first projects the original data set D onto the working set $\boldsymbol{S}$ to get a smaller data set D' . Then it obtains from D' the best LTMs m_1 and m_2 that contains only 1 latent variable or no more than 2 latent variables respectively. BI concludes that the *UD-test* passes if and only if one of the two conditions is satisfied :

(1) m_2 contains only one latent variable, or (2) m_2 contains two latent variables and

$$BIC(m_2 \mid D') - BIC(m_1 \mid D') \leq \delta, \tag{3}$$

where δ is a threshold parameter. The left hand side of this inequation is an approximation to the natural logarithm of the Bayes factor [10] for comparing model m_2 with model m_1.

To illustrate the process, suppose that the working set $\boldsymbol{S}$=$\{X_1$, $X_2\}$ initially. Then X_3 and X_4 were subsequently added to $\boldsymbol{S}$ and the UD-test passed on both cases. Now consider adding X_5. Assume the models m_1 and m_2 for the set $\boldsymbol{S}$=$\{X_1$, X_2, X_3, X_4, $X_5\}$ are as shown in Figure 2, and suppose the BIC score of m_2 exceeds that of m_1 by threshold δ. Then UD-test fails and BI stops growing the set $\boldsymbol{S}$.

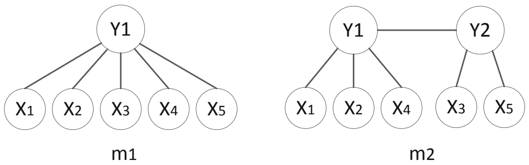

Fig. 2. The two models m_1 and m_2 considered in the UD-test

When the UD-test fails, model m_2 gives us two potential sibling clusters. If one of the two potential sibling clusters contains both the two initial variables, it is picked. Otherwise, BI picks the one with more variables and breaks ties arbitrarily. In the example, the two potential sibling clusters are $\{X_1, X_2, X_4\}$ and $\{X_3, X_5\}$. BI picks $\{X_1, X_2, X_4\}$ because it contains both the two initial variables X_1 and X_2.

After the first sibling cluster is determined, BI removes the variables in the cluster from the data set, and repeats the process to find other sibling clusters. This continues until all variables are grouped into sibling clusters.

After the sibling clusters are determined, BI introduces a latent variable for each sibling cluster. The cardinality of latent variable is automatically determined during the learning process. All the latent variables are further connected to form a tree structure by using Chow-Liu's algorithm [7]. At the end, BI carries out adjustments to the structure based on global considerations.

3.2 Discovering Co-occurrence of Patterns

As illustrated in Figure 1, BI yields flat latent tree models where the latent variables capture patterns of word co-occurrences. The patterns themselves may co-occur. Such higher level co-occurrence patterns can be discovered by recursively applying BI. This is what the HLTA algorithm is designed to do.

For reasons that will become clear later, we call the latent variables for word co-occurrence patterns level-1 latent variables. To discover higher level co-occurrence patterns, we first project the data onto the latent space spanned by the level-1 latent variables (Line 9). This is done by carrying out inference in the current model M_{whole}. For each data case d_i and each level-1 latent variable $Z1j$, we compute the posterior probability $P(Z1j|d_i, M_{whole})$, and assign the data case to the state with the maximum probability. In other words, we set the value of $Z1j$ to the state with highest

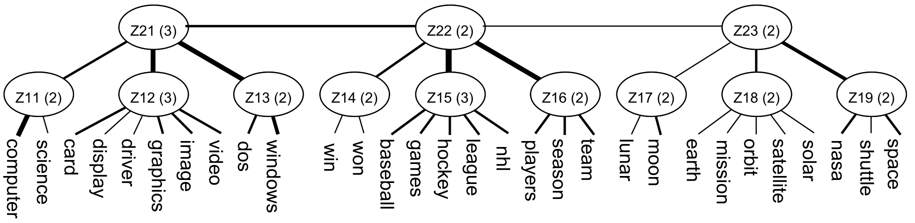

Fig. 3. The hierarchical latent tree model obtained by HLTA. The level-1 latent variables are found by running BI on the original data (*cf.* Figure 1) , and the level-2 latent variables are found by running BI on the data projected to the space spanned by the level-1 latent variables.

posterior probability. The values of the level-1 latent variables make up the projected data set D' (Line 9).

Next, we run BI algorithm on the projected data (Line 3). The resulting model is denoted as M in Algorithm 1. In this model, the level-1 latent variables are partitioned into sibling clusters and a level-2 latent variable is introduced for each cluster. The level-1 latent variables capture patterns of word co-occurrence, while level-2 variables capture the co-occurrence patterns of those patterns.

Now we have two flat LTMs, M_{whole} and M. M_{whole} consists of word variables and level-1 latent variables, and M consists of level-1 latent variables (viewed as observed variables) and level-2 latent variables. At the third step (Line 7), the two models are merged into a new LTM that consists of two levels of latent variables. Let us illustrate this using Figure 1. Assume the level-1 latent variables are partitioned into three clusters {Z11, Z12, Z13}, {Z14, Z15, Z16} and {Z17, Z18, Z19}, and suppose the corresponding level-2 latent variables are Z21, Z22 and Z23 respectively. Then, after merging M_{whole} and M at Line 7, we get the model shown in Figure 3. Basically, we stack M on top of M_{whole}, and then remove the connections among the level-1 latent variables. In the merged model M_{whole}, the parameters that involve the top level latent variables are estimated using Expectation-Maximization (EM) algorithm [9], and the values for other parameters are copied from the previous M_{whole} model.

After the level-2 latent variables are added to the model, we repeat the process to add more levels of latent variables until a predetermined number of levels k is reached, or when there are no more than two latent variables at the top level.

3.3 Topic Extraction

At the end of the while-loop (Line 11), HLTA has built a model with multiple levels of latent variables. The top level consists of either a single latent variable or multiple latent variables connected up in a tree structure. The other levels consist of multiple latent variables, each of which is connected to one latent variable at the level above and several variables at the level below. The bottom level consists of the word variables. We call the model a hierarchical latent tree model (HLTM). Each latent variable in the model represents a soft partition of the documents and its states can be interpreted as topics. In the last step (Line 12), HLTA computes descriptions of topics.

To see how the topics should be described, first consider the level-1 latent variable $Z15$ in Figure 3. It is directly connected to *baseball*, *games*, *hockey*, *league*, and *nhl*.

Table 1. Topics given by latent variable $Z22$ in Figure 3. The font sizes for probability values indicate their magnitude, while the font sizes of words indicate the discerning index.

Z22	S0 (87%)	team	0.01	players	0.01	baseball	0	season	0	hockey	0	games	0.02	league	0	nhl 0
	S1 (13%)	team	0.42	players	0.29	baseball	0.28	season	0.26	hockey	0.26	games	0.31	league	0.22	nhl 0.18

During data analysis, $Z15$ was introduced to model the correlations among those five word variables. Hence those words form the base for interpreting $Z15$. For level-2 latent variable $Z22$, it was introduced during data analysis to model the correlations among the level-1 latent variables $Z14$, $Z15$ and $Z16$. Hence all the words in the subtree rooted at $Z22$ form the base for its interpretation. We call the collection of those words the semantic base of $Z22$.

For a latent variable at a high level of the hierarchy, the semantic base can be large. To deal with the issue, we introduce the concept of *effective semantic base*. Sort all the word variables in the semantic base of a latent variable Z as X_1, X_2, $\cdots$, X_m in descending order of their mutual information $I(Z; X_i)$ with Z. Consider the mutual information $I(Z; X_1 \cdots X_i)$ between Z and the first i word variables. It monotonically increases with i, and reaches the maximum when $i = m$. The ratio $I(Z; X_1 \cdots X_i)/I(Z; X_1 \cdots X_m)$ is called the *information coverage* of the first i variables [4]. We define the *effective semantic base* of Z to be the collection of first i word variables for which the information coverage exceeds 0.95. Intuitively, $I(Z; X_1 \cdots X_m)$ is the amount of information about Z that is contained in its semantic base. The effective semantic base covers 95% of that information, hence is sufficient to determine the meaning of Z. The effective semantic bases for the level-2 latent variables in Figure 3 are shown in the following.

Z21	windows dos computer card graphics video image
Z22	team baseball players hockey season games league nhl
Z23	space nasa orbit shuttle mission earth moon solar

The meaning of a state of a latent variable is determined by the conditional distributions of the word variables from the effective semantic base. For example, the latent variable Z22 in Figure 3 has two states S0 and S1. The conditional probabilities (i.e., P(word=1|Z22=Si)) are given in Table 1. We see that in the cluster Z22=S1, the words *team*, *players*, *baseball* etc. occur with relatively high probabilities. It can be interpreted as the topic *sports*. On the other hand, the words seldom occur in cluster Z22=S0 which can be considered as a background topic. Those two topics consist of 13% (P(Z22=S1)=0.13) and 87% (P(Z22=S0)=0.87) of the documents respectively.

To highlight the importance of words in characterizing a topic, we introduce the concept of *discerning index*. Let Z be a latent variable that has two or more states, and W be a word variable. For a given state s of Z, let Z_s be another variable that takes two possible values 0 and 1, with Z_s=1 meaning Z=s and Z_s=0 meaning $Z \neq s$. The discerning index of W for Z=s is the mutual information $I(W, Z_s)$ between W and Z_s. The higher the index, the more important W is for distinguishing the cluster Z=s from other clusters in the partition given by Z. Usually, the words that occur with high probabilities in Z=s and low probabilities when $Z \neq s$ have high discerning index values. In Table 1, the order and font sizes of the words are determined according to the discerning index. The font sizes for the probability values are determined by their own magnitude. This visualization scheme is proposed so that the thematic meaning of each topic is readily visible to the reader.

At line 12, HLTA computes the effective semantic base of each latent variable and, for each topic, the discerning index and occurrence probability of each word from the base. The probability of each topic in the entire corpus and the probability of each document belonging to each topic are also computed. All the computations need to perform inference in the resulting model, which can be done in HLTM by applying standard algorithms, e.g. clique tree propagation [18].

4 Empirical Results

In this section, we demonstrate the characteristics of HLTA topics and the topic hierarchy. We also compare the predictive performance of HLTA on several data sets which include: (1) NIPS[1] data: 1,740 NIPS articles published from 1988 to 1999; (2) JACM[2] data: 536 abstracts from the Journal of the ACM; (3) Newsgroup[3] data: about 20,000 newsgroup documents. For JACM data, all 1,809 words were used in the experiments. For Newsgroup and NIPS data, the vocabulary was restricted to 1,000. The stop words and words appear in less than ten papers were removed. Then we computed the TF-IDF value of each word in each document. The top 1,000 words with highest average TF-IDF value were selected. The code and data sets used in the experiment are available at: `http://www.cse.ust.hk/~lzhang/ltm/index.htm`.

4.1 Results on the NIPS Data

We first show the results of HLTA on NIPS data.

Model Structure. The analysis resulted in a hierarchical LTM with 382 latent variables arranged in 5 levels. There are 279, 72, 21, 8 and 2 latent variables on levels 1, 2, 3, 4 and 5 respectively. Table 2 shows part of the hierarchical structure. Table representation is used instead of the tree structure to save space. We see that, for example, the word variables *bayesian*, *posterior*, *prior*, *bayes*, *priors*, *framework*, *gamma* and *normal* are connected to the level-1 latent variable Z106; the level-1 latent variables Z106-Z112 are connected to the level-2 latent variable Z203; the level-2 latent variables Z201-Z205 are connected to the level-3 latent variable Z301; the level-3 latent variables Z301-Z302 are connected to a latent variable at level 4 (which is not shown); and so on.

The words are displayed in different font sizes to indicate their mutual information with the level-1 latent variables to which they are connected. For example, Z106 is more strongly correlated with *bayesian* than *normal*. The names of the latent variables are also displayed in different font sizes to indicate their mutual information with the parent latent variables at the next higher level.

[1] `http://www.cs.nyu.edu/~roweis/data.html`

[2] `http://www.cs.princeton.edu/~blei/downloads/`

[3] `http://qwone.com/~jason/20Newsgroups/`

Table 2. Part of the hierarchical latent tree model obtained by HLTA on the NIPS data

Z301(2)	Z201(2)	Z101(2): likelihood conditional log em maximum ix derived mi ; Z102(2): statistical statistics ; Z103(2): density densities
	Z202(2)	Z104(2): entropy divergence mutual ; Z105(2): variables variable
	Z203(3)	Z106(3): bayesian posterior prior priors bayes framework gamma normal ; Z107(2): probabilistic distributions probabilities ; Z108(2): inference gibbs sampling generative uncertainty ; Z109(2): mackay independent averaging ensemble uniform ; Z110(2): belief graphical variational ; Z111(2): monte carlo ; Z112(2): uk ac generalisation
	Z204(3)	Z113(2): mixture mixtures latent fit ; Z114(3): multiple hierarchical individual sparse missing multi significant index represent hme ; Z115(2): experts expert gating ; Z116(2): weighted sum weighting ; Z117(2): scale scales scaling
	Z205(3)	Z118(3): estimate estimation estimated estimates estimating measure deviation ; Z119(2): estimator true unknown ; Z120(2): sample samples ; Z121(2): assumption assume assumptions assumed ; Z122(2): observations observation observed ; Z123(2): computed compute
Z302(3)	Z206(4)	Z124(2): gaussian covariance variance gaussians program provided ; Z125(2): subspace dimensionality orthogonal reduction ; Z126(3): component components principal pca decomposition ; Z127(2): dimension dimensional dimensions vectors ; Z128(2): matrix matrices diagonal ; Z129(2): exp cr exponential ; Z130(2): noise noisy robust ; Z131(2): projection projections pursuit operator ; Z132(2): radial basis rbf ; Z133(2): column row ; Z134(2): eq fig proc
	Z207(2)	Z135(2): eigenvalues eigenvalue eigenvectors identical ; Z136(2): ij product wij bi ; Z137(2): modes mode
	Z208(2)	Z138(2): mixing coefficients inverse joint smooth smoothing ; Z139(2): blind ica separation sejnowski natural concept ; Z140(2): sources source
Z303(2)	Z209(2)	Z141(2): classification classifier classifiers nn ; Z142(2): class classes
	Z210(2)	Z143(2): discriminant discrimination fisher ; Z144(2): labels label labeled
	Z211(2)	Z145(2): handwritten digit digits le ; Z146(2): character characters handwriting
Z304(3)	Z212(3)	Z147(2): regression regularization generalization risk ; Z148(3): vapnik svm margin support vc dual fraction ; Z149(2): kernel kernels ; Z150(2): empirical drawn theoretical ; Z151(2): xi yi xj zi xl gi
	Z213(2)	Z152(2): validation cross bias ; Z153(2): stopping pruning criterion obs ; Z154(2): prediction predictions predict predicted predictive
	Z214(2)	Z155(2): machines machine boltzmann ; Z156(2): boosting adaboost weak

We can see from Table 2 that many level-1 latent variables represent thematically meaningful patterns. Examples include Z101 (*likelihood conditional log etc.*), Z106 (*Bayesian posterior prior etc.*), Z108 (*inference gibbs sampling etc.*), Z111 (*monte carlo*), Z124 (*gaussian covariance variance etc.*), Z125 (*subspace dimensionality orthogonal reduction*), Z139 (*blind ica separation*), Z145 (*handwritten digit*), Z148 (*vapnik svm margin support etc.*), Z155 (*machines Boltzmann*), Z156 (*boosting adaboost*).

For latent variables at level 2 and level 3, their effective semantic bases are given in Table 3. For latent variables at different levels, a higher level latent variable represents a partition of documents based on a wider selection of words than its children. It is usually about a general concept that has several aspects. We see in Table 3 that Z301 is about probabilistic method, while its children cover likelihood (Z201), entropy (Z202), Bayesian (Z203), mixture (Z204) and estimate (Z205). Z302 is about the use of Gaussian covariance matrix, while its children cover PCA (Z206), eigenvalue/vector (Z207), and blind source separation (Z208); Z303 is about classification, while its

Table 3. Effective semantic bases of level-2 and level-3 latent variables

Z301		likelihood bayesian statistical conditional posterior probabilistic density log mixture prior bayes distributions estimate priors
	Z201	likelihood statistical conditional density log em statistics
	Z202	entropy variables variable divergence
	Z203	bayesian posterior probabilistic prior bayes distributions priors inference monte carlo probabilities
	Z204	mixture mixtures experts hierarchical latent expert weighted sparse
	Z205	estimate estimation estimated estimates estimating estimator sample true samples observations
Z302		gaussian covariance matrix variance eigenvalues eigenvalue exp gaussians pca principal matrices eigenvectors component noise
	Z206	gaussian matrix covariance pca variance principal subspace dimensionality projection exp gaussians
	Z207	eigenvalues eigenvalue eigenvectors
	Z208	blind mixing ica coefficients inverse separation sources joint
Z303		classification classifier classifiers class classes handwritten discriminant digit
	Z209	classification classifier classifiers class
	Z210	discriminant label labels discrimination
	Z211	handwritten digit character digits characters
Z304		regression validation vapnik svm machines regularization margin generalization boosting kernel kernels risk empirical
	Z212	regression vapnik svm margin kernel regularization generalization kernels support xi risk
	Z213	validation cross stopping pruning prediction predictions
	Z214	machines boosting machine boltzmann

children cover discriminant and handwritten digit/character recognition. Z304 is about regression, while its children cover SVM, cross validation, and boosting.

Topics. Each latent variable in the model represents a soft partition of the documents. Each state of the latent variables corresponds to a cluster in the partition and can be interpreted as a topic. As discussed in Section 3.3, we characterize the topic using words from its effective semantic base with the highest discerning indices, that is, the words that best distinguish the documents in the cluster from documents not in the cluster. For non-background topics, those usually are the words that appear with high probability in the cluster and low probability in other clusters.

Take Z301 as an example. It has two states and hence partitions the documents into two clusters. The characterizations of the two clusters are given in Table 4. We see that, for Z301=S1, the words *likelihood*, *Bayesian* and *statistical* are placed at the beginning of the list. They have the highest discerning indices. Their probabilities of occurring in Z301=S1 are 0.58, 0.45 and 0.73 respectively, which are significantly larger than those for Z301=S0, which are 0.06, 0.03, and 0.24 respectively. On the other hand, the word *estimate* also occurs with high probability (0.64) in Z301=S1. However, it has low discerning index for Z301=S1 because its probability in the other cluster Z301=S0 is also relatively high (0.25). It is clear from the characterization that the topic Z301=S1 is about general probabilistic method, while Z301=S0 is a background topic. They consist of 34% and 66% of the documents respectively.

Table 4 shows characterizations of topics given by ten latent variables. The three level-3 latent variables (i.e., Z301, Z303 and Z304) are chosen because they are about the basic topics covered in a typical machine learning course, namely probabilistic methods, classification and regression. The others are selected level-2 and level-1

Table 4. Example topics found by HLTA on NIPS data. For each topic, only the words in the effective semantic base are shown. The order and font sizes of words in each topic are determined by discerning index. The font sizes of word occurrence probabilities simply reflect their magnitude.

Z301	S0 (66%)	likelihood 0.06 bayesian 0.03 statistical 0.24 conditional 0.05 posterior 0.04 density 0.13 probabilistic 0.06 log 0.18 bayes 0.02 mixture 0.06 prior 0.15 estimate 0.25 distributions 0.15 priors 0.01
	S1 (34%)	likelihood 0.58 bayesian 0.45 statistical 0.73 conditional 0.4 posterior 0.37 density 0.52 probabilistic 0.42 log 0.58 bayes 0.29 mixture 0.4 prior 0.54 estimate 0.64 distributions 0.52 priors 0.22
Z203	S1 (19%)	probabilistic 0.48 distributions 0.58 probabilities 0.49 bayesian 0.4 prior 0.52 posterior 0.32 bayes 0.26 priors 0.17 inference 0.22 carlo 0.05 monte 0.05
	S2 (11%)	bayesian 0.75 monte 0.54 carlo 0.53 posterior 0.64 inference 0.58 prior 0.79 priors 0.41 bayes 0.47 probabilistic 0.54 distributions 0.63 probabilities 0.54
Z113	S1 (19%)	mixture 0.76 mixtures 0.53 latent 0.17 fit 0.34
Z303	S1 (30%)	classification 0.81 classifier 0.48 classifiers 0.38 class 0.69 classes 0.53 handwritten 0.22 discriminant 0.15 digit 0.2
Z211	S1 (13%)	handwritten 0.58 digit 0.52 character 0.54 digits 0.42 characters 0.31
Z145	S1 (12%)	handwritten 0.72 digit 0.64 digits 0.52
Z146	S1 (9%)	character 0.84 characters 0.49 handwriting 0.24
Z304	S1 (25%)	regression 0.36 validation 0.33 regularization 0.21 generalization 0.5 risk 0.15 empirical 0.31 svm 0 boosting 0 machines 0.1 vapnik 0.09 margin 0.04 kernels 0.07 kernel 0.11
	S2 (7%)	machines 0.76 svm 0.42 vapnik 0.53 margin 0.44 boosting 0.3 kernel 0.55 kernels 0.4 regression 0.49 validation 0.42 generalization 0.62 regularization 0.29 empirical 0.44 risk 0.21
Z213	S1 (19%)	validation 0.57 cross 0.55 stopping 0.24 pruning 0.18 prediction 0.48 predictions 0.35
Z156	S1 (3%)	boosting 0.9 adaboost 0.35 weak 0.39

latent variables under the level-3 latent variables. The background topics are not shown except Z301=S0. These topics show clear thematic meaning. We can see that Z301=S1 is about probabilistic method in general, while its subtopics Z203=S2 is about Bayesian-monte-carlo, Z203=S1 is about probabilistic method not involving monte-carlo, Z113=S1 is about mixture models. Topic Z303=S1 is about classification, while its subtopics Z211=S1 is about digit/character classification, Z145=S1 is about handwritten digit classification, Z146=S1 is about handwritten character classification. Z304=S1 is about regression in general, while its subtopics Z304=S2 is about SVM, Z213=S1 is about cross validation and Z156=S1 is about boosting.

Comparisons with LDA. To better appreciate the topics found by HLTA, it is necessary to compare them with those detected by the LDA approach [3]. In this section, we run LDA on the NIPS data to find 150 topics. The documents are represented as bags-of-words in LDA, while as binary vectors in HLTA.

Table 5 shows the LDA topics that are the closest in meaning to the HLTA topics shown in Table 4. They are selected using the top three words of the HLTA topics. The LDA topic that best matches the HLTA topic is selected manually. The LDA approach produces flat topics. It does not organize the topics in a hierarchical structure as in HLTA. Thus in this section, we focus on the topics. Compared the HLTA topics with the LDA topics, we can find that they differ in two fundamental ways. First, an HLTA topic corresponds to a collection of documents. As such, we can talk about the size of a topic, which is the fraction of documents belonging to the topic among all documents.

Table 5. LDA topics that correspond to the HLTA topics in Table 4. Only the top eight words are shown for each topic.

HLTA topic	LDA topic
Z301=S1	T-25: bayesian .16 prior .13 posterior .10 evidence .04 bayes .04 priors .03 log .03 likelihood .03
Z203=S2	T-97: gaussian .23 monte .07 carlo .07 covariance .05 variance .05 processes .04 williams .03 exp .02
Z113=S1	T-78: mixture .13 em .12 likelihood .11 gaussian .05 log .04 maximum .04 mixtures .04 latent .03
Z303=S1	T-139: classification .20 class .19 classifier .15 classifiers .09 classes .09 decision .03 bayes .02 labels .01
Z211=S1	T-120: recognition .18 character .09 digit .06 characters .06 digits .05 handwritten .05 segmentation .04 le .02
Z304=S1	T-84: regression .15 risk .08 variance .08 bias .08 confidence .04 empirical .03 smoothing .03 squared .03
Z304=S2	T-141: kernel .16 support .09 svm .05 kernels .05 machines .04 margin .03 vapnik .02 feature .02
Z213=S1	T-146: cross .21 validation .19 stopping .07 generalization .05 selection .04 early .04 fit .02 statistics .02
Z156=S1	T-45: margin .09 hypothesis .08 weak .07 boosting .06 generalization .05 adaboost .04 algorithms .03 base .02

In Table 4, the numbers shown in parenthesis indicate the size of each HLTA topic. On the other hand, LDA treats each document as a mixture of topics. It is possible to aggregate the topic proportions of all documents. However, the aggregated quantity would be the fraction of words belonging to that topic, not the fraction of documents.

A more important difference lies in the way topics are characterized. When picking words to characterize a topic, HLTA uses discerning index which considers two factors: (1) the word occurrence probability in documents belonging to the topic, and (2) the word occurrence probability in documents not on the topic. This results in clear and clean topic characterizations. The consideration of the second factor implies that HLTA is unlikely to pick polysemous words when characterizing a topic, because such words are used in multiple topics. This should not affect topic identification as long as there are words peculiar to each topic. In contrast, LDA considers only the first factor, and the resulting topic characterizations are sometimes not clear.

For example, we first look at Z113 which has two states. According to Table 4, the words *mixture*, *mixtures* and *latent* have the highest discerning indices in Z113=S1, which indicate they occur with high probability in Z113=S1 and low probability in background topic (i.e., Z113=S0). Z113=S1 is clearly about mixture models. The closest LDA topic is Topic T-78. As shown in Table 5, the leadings words in the topic are *mixture*, *em* and *likelihood*. The words *em* and *likelihood* are not characteristic of mixture models because they are used more often in other situations such as the handling of missing data. The HLTA characterization seems cleaner.

For Z156 in Table 4, the words with highest discerning indices in Z156=S1 are *boosting*, *adaboost* and *weak*, which occur with high probability in Z156=S1 and low probability in background topic (i.e., Z156=S0). Z156=S1 is clearly about boosting. The closest LDA topic is Topic T-45. As shown in Table 5, the leadings words in the topic are *margin*, *hypothesis*, *weak*, *generalization* and *boosting*. It is not clear to us at the first sight what this topic is about.

Comparisons with HLDA. The HLDA [2] approach, which is an extension of LDA, can also learn topic hierarchies from data. To compare HLTA with HLDA, we trained a three-level HLDA[4] on NIPS data. The hyperparameters of HLDA are set according to

[4] The code of HLDA is obtained from: `http://www.cs.princeton.edu/~blei/topicmodeling.html`

Table 6. Part of the topic hierarchy produced by HLDA on NIPS data. Only the top ten words are shown for each topic.

[Topic L3]	units hidden layer unit weight test noise inputs trained patterns
[Topic L2-0]	gaussian likelihood density log mixture em prior posterior estimate estimation
[Topic L1-0]	kernel xi pca kernels feature regression matrix support svm principal
[Topic L1-1]	evidence bayesian gaussian posterior prior approximation field mackay variational exp
[Topic L1-2]	validation cross generalization stopping variance examples early prediction estimator penalty
[Topic L1-3]	sampling carlo bayesian monte prior predictive posterior inputs priors loss
[Topic L1-4]	class bayes matrix coding learned max nearest classes classifier classifiers
[Topic L1-5]	propagation belief inference jordan nodes tree variational variables product graphical
[Topic L2-1]	recognition image feature block le images address features handwritten digit
[Topic L1-6]	characters character recognition net field segmentation fields word digits window
[Topic L1-7]	image images digits digit transformation convex pixel generative object control
[Topic L2-2]	regression image classification representation mixture prediction capacity selection weight classifier
[Topic L1-8]	adaboost cost boosting margin potential algorithms ct hypothesis base weak
[Topic L1-9]	transform pca dimension coding reduction mixture image images reconstruction grid

the settings used in [2]. Table 6 shows part of the topic hierarchy. Only the root topic (i.e., Topic-L3) and the topics that match the HLTA topics in Table 4 are presented.

A comparison of Tables 4 and 6 suggests that the thematic meaning of the topic hierarchy given by HLDA is not as clear as that given by HLTA. For example, we can first look at Topic L2-0 in Table 6. The top words of Topic L2-0 are *gaussian*, *likelihood*, *density* and *log*. It can be interpreted as a topic about probabilistic methods. Most subtopics of Topic L2-0 are about probabilistic methods, e.g., Topic L1-1 (evidence-bayesian-gaussian-posterior) and Topic L1-3 (sampling-carlo-bayesian -monte). However, there are also subtopics that are not about probabilistic methods. In particular, Topic L1-2 (validation-cross-generalization-stopping) is about cross validation. It can hardly be viewed as a subtopic of probabilistic methods. In contrast, all the subtopics of the HLTA topic Z301=S1 are about probabilistic methods.

As another example, consider the HLDA topics Topic L1-6 (characters-character -recognition-net) and Topic L1-7 (image-images-digits-digits). They can be interpreted as “character recognition” and “digit recognition” respectively. However, the meaning of their parent topic, i.e., Topic L2-1 (recognition-image-feature-block), is not clear to us. The HLTA topics Z145=S1 (handwritten-digit-digits) and Z146=S1 (character -characters-handwriting), as shown in Table 4, are about the same topics as Topic L1-7 and Topic L1-6. They seem to give better characterizations of the topics. More importantly, they are subtopics of Z303=S1 (classification-classifier-classifiers-class) and Z211=S1 (handwritten-digit-character-digits), which is clearly reasonable.

In summary, in HLTA, the topics at higher level are more general topics since they are defined on a larger semantic base. A subtopic, on the other hand, is defined on a subset of the semantic base of its parent topic. Thus the subtopics in HLTA are more specific topics. They are semantically close to their parent topics since they share part of the semantic base. Compared to HLDA topics in Table 6, the parent topic and child topics in HLTA, as shown in Table 4, show higher semantic closeness.

Topic Semantic Coherence. To quantitatively compare the quality of topics found by LDA, HLDA and HLTA on NIPS data, we compute their *topic coherence scores* [14]. The topic coherence score for topic t is defined as

$$C(t, W^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(w_m^{(t)}, w_l^{(t)}) + 1}{D(w_l^{(t)})}, \tag{4}$$

where $W^{(t)} = \{w_1^{(t)}, ..., w_M^{(t)}\}$ are the top M words used to characterize topic t, $D(w_i)$ is the document frequency of word w_i and $D(w_i, w_j)$ is the co-document frequency of words w_i and w_j. Document frequency is the number of documents containing the words. Given two collections of topics, the one with higher average topic coherence is regarded as better. For comparability, all topics from the two collections should be of the same length, that is, characterized by the same number (i.e., M) of words.

For HLTA, it produced 140 non-background topics by latent variables from levels 2, 3 and 4. We consider two scenarios in terms of the number of words we use to characterize the topics. In the first scenario, we set M=10. The level-2 topics are excluded in this scenario because the semantic bases of some level-2 latent variables consist of fewer than 10 words. As a result, there are only 47 topics. In the second scenario, we set M=4. Here all the 140 topics are included. LDA was instructed to find 47 and 140 topics for the two scenarios respectively. HLDA produced 179 topics. For comparability, 47 and 140 topics were sampled for the two scenarios respectively.

The average coherence of the topics produced by the three methods are given in Table 7. we see that the value for HLTA is significantly higher than those for LDA and HLDA in both scenarios. The statistics suggest that HLTA has found, on average, thematically more coherent topics than LDA and HLDA.

Table 7. Average topic coherence for topics found by LDA, HLDA and HLTA on NIPS data

	M	Number of topics	Avg. Coherence
HLTA(L3-L4)	10	47	-47.26
LDA	10	47	-55.38
HLDA-s	10	47	-62.83
HLDA	10	179	-63.32
HLTA(L2-L3-L4)	4	140	-5.89
LDA	4	140	-7.81
HLDA-s	4	140	-7.97
HLDA	4	179	-7.98

4.2 Likelihood Comparison

Having compared HLTA, LDA and HLDA in terms of the topics they produce, we next compare them as methods for text modeling. The comparison is in terms of per-document held-out log likelihood. For compatibility, LDA was run on both the count data and the binary version data. The results on the count data are denoted as LDA-C-100. For the binary data, several possibilities were tried for the number of topics, namely 20, 40, 60 and 80. For HLDA, the hyperparameters are set according to the settings used in [2]. For HLTA, three possibilities were tried for the UD-test threshold δ, namely 1, 3 and 5, which are suggested by [10].

Table 8. Per-document held-out log likelihood for HLTA, HLDA and LDA on test data. Results are averaged over five-fold cross validation.

	JACM	NIPS	NEWSGROUP
HLTA-1	-226±6	-394± 3	-113±1
HLTA-3	-226±6	-394± 3	-113±1
HLTA-5	-226±6	-394± 3	-113±1
HLDA	-220±39	-529±23	-98±8
LDA-20	-489±20	-1229±18	-197±2
LDA-40	-498±20	-1240±17	-199±2
LDA-60	-505±20	-1250±18	-199±2
LDA-80	-510±21	-1257±18	-199±2
LDA-C-100	-819±37	-3413± 35	-289±6

The results are given in Table 8. They show that HTLA and HLDA performed much better than LDA on all the data sets in the sense that the models they produced are much better in predicting unseen data than those obtained by LDA. The differences become larger when LDA was run on count data (last row of Table 8) rather than binary data. HLTA is better than HLDA on the NIPS data. However, HLDA is slightly better than HLTA on the other two data sets. In terms of running time, HLDA and HLTA are significantly slower than LDA. For example, for the binary version NIPS data, LDA took about 3.5 hours, while HLTA and HLDA took about 17 hours and 68 hours respectively.

5 Related Work

There are some other methods for learning latent tree models. We refer the readers to [15] for a detailed survey. Most of the methods are designed for density estimation [6], latent structure discovery [5] and multi-dimensional clustering [4]. None of these methods are designed for topic detection. More importantly, these methods do not provide a principled way to extract the topics from the model when they are applied on text data.

HLTM also resembles hierarchical clustering. However, there are fundamental differences between the two models. First, an HLTM is a probabilistic graphical model which allows inference among variables, while the structure given by traditional hierarchical clustering is not. Second, an HLTM can be also seen as a clustering tool which clusters the data points and variables simultaneously, while hierarchical clustering can only be used to cluster either data points or variables.

6 Conclusions and Future Directions

We propose a new method called HLTA for topic detection. The idea is to model patterns of word co-occurrence and co-occurrence of those patterns using a hierarchical latent tree model. Each latent variable in HLTM represents a soft partition of documents. The document clusters in each partition are interpreted as topics. Each topic is characterized using the words that occur with high probability in documents belonging to the topic and occur with low probability in documents not belonging to the topic. Empirical results indicate that HLTA can identify rich and thematically meaningful topics of various generality. In addition, HLTA can determine the number of topics automatically and organize topics into a hierarchy. Currently, HLTA treats words as binary variables. One future direction is to extend it so that it can handle count data. A second direction is to develop faster algorithms for learning hierarchical latent tree models.

Acknowledgments. We thank the anonymous reviewers for their valuable comments. Research on this paper was supported by Guangzhou HKUST Fok Ying Tung Research Institute.

References

1. Bartholomew, D., Knott, M., Moustaki, I.: Latent Variable Models and Factor Analysis. A Unified Approach. John Wiley & Sons (2011)
2. Blei, D., Griffiths, T., Jordan, M.: The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. J. ACM 57(2), 7:1–7:30 (2010)
3. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. the Journal of Machine Learning Research 3, 993–1022 (2003)
4. Chen, T., Zhang, N.L., Liu, T.F., Poon, K.M., Wang, Y.: Model-based multidimensional clustering of categorical data. Artif. Intell. 176(1), 2246–2269 (2012)
5. Chen, T., Zhang, N.L., Wang, Y.: Efficient model evaluation in the search-based approach to latent structure discovery. In: 4th European Workshop on Probabilistic Graphical Models, pp. 57–64 (2008)
6. Choi, N.J., Tan, V.Y.F., Anandkumar, A., Willsky, A.: Learning latent tree graphical models. Journal of Machine Learning Research 12, 1771–1812 (2011)
7. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14(3), 462–467 (1968)
8. Cover, T., Thomas, J.: Elements of Information Theory. Wiley-Interscience (2006)
9. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. J. Royal Statistical Society, Series B 39(1), 1–38 (1977)
10. Kass, R., Raftery, A.: Bayes factor. Journal of American Statistical Association 90(430), 773–795 (1995)
11. Lafferty, J., Blei, D.: Correlated topic models. In: NIPS, pp. 147–155 (2005)
12. Liu, T.F., Zhang, N.L., Chen, P., Liu, H., Poon, K.M., Wang, Y.: Greedy learning of latent tree models for multidimensional clustering. Machine Learning (2013), doi: 10.1007/s10994-013-5393-0
13. Liu, T.F., Zhang, N.L., Poon, K.M., Liu, H., Wang, Y.: A novel ltm-based method for multi-partition clustering. In: 6th European Workshop on Probabilistic Graphical Models, pp. 203–210 (2012)
14. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: EMNLP, pp. 262–272 (2011)
15. Mourad, R., Sinoquet, C., Zhang, N.L., Liu, T.F., Leray, P.: A survey on latent tree models and applications. J. Artif. Intell. Res. (JAIR) 47, 157–203 (2013)
16. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc. (1988)

17. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics 6, 461–464 (1978)
18. Shafer, G., Shenoy, P.: Probability propagation. Annals of Mathematics and Artificial Intelligence 2(1-4), 327–351 (1990)
19. Zhang, N.L.: Hierarchical latent class models for cluster analysis. Journal of Machine Learning Research 5(6), 697–723 (2004)