# Tracking Dynamics of Topic Trends Using a Finite Mixture Model

Satoshi Morinaga
NEC Corporation
4-1-1,Miyazaki,Miyamae,
Kawasaki,Kanagawa 216-8555,JAPAN
morinaga@ccm.cl.nec.co.jp

Kenji Yamanishi
NEC Corporation
4-1-1,Miyazaki,Miyamae,
Kawasaki,Kanagawa 216-8555,JAPAN
k-yamanishi@cw.jp.nec.com

## ABSTRACT

In a wide range of business areas dealing with text data streams, including CRM, knowledge management, and Web monitoring services, it is an important issue to discover topic trends and analyze their dynamics in real-time. Specifically we consider the following three tasks in topic trend analysis: 1) *Topic Structure Identification*; identifying what kinds of main topics exist and how important they are, 2) *Topic Emergence Detection*; detecting the emergence of a new topic and recognizing how it grows, 3) *Topic Characterization*; identifying the characteristics for each of main topics. For real topic analysis systems, we may require that these three tasks be performed in an on-line fashion rather than in a retrospective way, and be dealt with in a single framework. This paper proposes a new topic analysis framework which satisfies this requirement from a unifying viewpoint that a topic structure is modeled using a finite mixture model and that any change of a topic trend is tracked by learning the finite mixture model dynamically. In this framework we propose the usage of a time-stamp based discounting learning algorithm in order to realize real-time topic structure identification. This enables tracking the topic structure adaptively by forgetting out-of-date statistics. Further we apply the theory of dynamic model selection to detecting changes of main components in the finite mixture model in order to realize topic emergence detection. We demonstrate the effectiveness of our framework using real data collected at a help desk to show that we are able to track dynamics of topic trends in a timely fashion.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

## Keywords

topic analysis, model selection, CRM, text mining


*KDD'04,* August 22–25, 2004, Seattle, Washington, USA.


## 1. INTRODUCTION

### 1.1 Problem Setting

In a wide range of business areas dealing with text streams, including CRM, knowledge management, and Web monitoring services, it is an important issue to discover topic trends and analyze their dynamics in *real-time*. For example, it is desired in the CRM area to grasp a new trend of topics in customers' claims every day and to track a new topic as soon as it emerges. A topic is here defined as a seminal event or activity. Specifically we consider the following three tasks in topic analysis:

1) *Topic Structure Identification*; learning a *topic structure* in a text stream, in other words, identifying what kinds of main topics exist and how important they are.
2) *Topic Emergence Detection*; detecting the emergence of a new topic and recognizing how rapidly it grows, similarly, detecting the disappearance of an existing topic.
3) *Topic Characterization*; identifying the characteristics for each of main topics.

For real topic analysis systems, we may require that these three tasks be performed in an on-line fashion rather than in a retrospective way, and be dealt with in a single framework.

The main purpose of this paper is to propose a new topic analysis framework that satisfies the requirement as above, and to demonstrate its effectiveness through its experimental evaluations for real data sets.

Our framework is designed from a unifying viewpoint that a topic structure in a text stream is modeled using a finite mixture model (a model of the form of a weighted average of a number of probabilistic models) and that any change of a topic trend is tracked by learning the finite mixture model dynamically. Here each topic corresponds to a single mixture component in the model.

All of the tasks 1)-3) are formalized in terms of a finite mixture model as follows: As for the task 1), the topic structure is identified by statistical parameters of a finite mixture model. They are learned using our original *time-stamp based discounting learning algorithm*, which incrementally and adaptively estimates statistical parameters of the model by gradually forgetting out-of-date statistics, making use of time-stamps of data. This makes the learning procedure adaptive to changes of the nature of text streams.

As for the task 2), any change of a topic structure is recognized by tracking the change of main components in a mixture model. We apply the theory of *dynamic model selection* [7] to detecting changes of the optimal number of

main components and their organization in the finite mixture model. We may recognize that a new topic has emerged if a new mixture component is detected in the model and remains for a while. Unlike conventional approaches to statistical model selection under the stationary environment, dynamic model selection is performed under the non-stationary one in which the optimal model may change over time. Further note that we deal with a complicated situation where the dimension of input data, i.e., the number of features of a text vector, may increase as time goes by.

As for the task 3), we classify every text into the cluster for which the posterior probability is largest, and then we characterize each topic using feature terms characterizing texts classified into its corresponding cluster. These feature terms are extracted as those of highest information gain, which are computed in real-time.

We demonstrate the validity of the topic trend analysis framework, by showing experimental results on its applications to real domains. Specifically we emphasize that it is really effective for discovering trends in questions at a help desk.

## 1.2 Related Work

The technologies similar to 1)-3) have extensively been explored in the area of *topic detection and tracking* (TDT) (see [1]). Actually 1) and 2) are closely related to the subproblems in TDT called *topic tracking* and *new event detection*, respectively. Here topic tracking is to classify texts into one of topics specified by a user, while new event detection, formerly called *first story detection*, is to identify texts that discuss a topic that has not already been reported in earlier texts. The latter problem is also related to work on topic-conditioned novelty detection by Yang et.al.[16]. In most of related TDT works, however, topic tracking or new event detection is conducted without identifying main topics or a topic structure, hence the tasks 1)-3) cannot be unified within a conventional TDT framework. Further topic timeline analysis has not been addressed in it.

Swan and Allen [12] addressed the issue of how to automatically overview timelines of a set of news stories. They used the $\chi^2$-method to identify at each time a burst of feature terms that more frequently appear than at other times. Similar issues are addressed in the visualization community [3]. However, all of the methods proposed there are not designed to perform in an on-line fashion.

Kleinberg [4] proposed a formal model of "bursts of activity" using an infinite-state automaton. This is closely related to topic emergence detection in our framework. A burst has a somewhat different meaning from a topic in the sense that the former is a series of texts including a specific feature, while the latter is a cluster of categorized texts. Hence topic structure identification and characterization cannot be dealt with in his model. Further note that Kleinberg's model is not designed for real-time analysis but for retrospective one.

Related to our statistical modeling of a topic structure, Liu et.al. [2] and Li and Yamanishi [6] also proposed methods for topic analysis using a finite mixture model. Specifically, Liu et.al. considered the problem of selecting the optimal number of mixture components in the context of text clustering. In their approach a single model is selected as an optimal model under the assumption that the optimal model does not change over time. Meanwhile, in our approach, a sequence of optimal models is selected dynamically under the assumption that the optimal model may change over time.

Related to topic emergence detection, Matsunaga and Yamanishi [7] proposed a basic method of dynamic model selection, by which one can dynamically track the change of number of components in the mixture model. However, any of all of these technologies cannot straightforwardly be applied to real-time topic analysis in which the dimension of data may increase as time goes by.

Related to topic structure identification, an on-line discounting learning algorithm for estimating parameters in a finite mixture model has been proposed by Yamanishi et. al. [14]. The main difference between our algorithm and theirs is that the former makes use of time-stamps in order to make the topic structure affected by a timeline of topics while the latter considers only the time-order of data ignoring their time-stamps.

The rest of this paper is organized as follows: Section 2 describes a basic model of topic structure. Section 3 gives a method for topic structure identification. Section 4 gives a method for topic emergence detection. Section 5 gives a method for topic characterization. Section 6 gives experimental results. Section 7 gives concluding remarks.

# 2. MODEL

We employ a probabilistic model called a *finite mixture model* for the representation of topic generation in a text stream. Let $\mathcal{W} = \{w_1, \cdots, w_d\}$ be the complete vocabulary set of the document corpus after the stop-words removal and words stemming operations. For a given document $x$, let $tf(w_i)$ be the term frequency of word $w_i$ in $x$. Let $idf(w_i)$ be the idf value of $w_i$, i.e., $idf(w_i) = \log(N/df(w_i))$ where $N$ is the total number of texts for reference and $df(w_i)$ is the frequency of texts in which $w_i$ appears. Let $tf\text{-}idf(w_i)$ be the tf-idf value of $w_i$ in $x$, i.e., $tf\text{-}idf(w_i) = tf(w_i) \times \log(N/df(w_i))$. We may represent a text $x$ of the form:

$$x = (tf(w_1), ..., tf(w_d))$$

or

$$x = (tf\text{-}idf(w_1), ..., tf\text{-}idf(w_d)).$$

We may use either type of the representation forms.

Let $K$ be a given positive integer representing the number of different topics. We suppose that a text has only one topic and a text having the $i$-th topic is distributed according to the probability distribution with a density: $p_i(x|\theta_i)$ $(i = 1, 2, \cdots, K)$, where $\theta_i$ is a real-valued parameter vector. We suppose here that $x$ is distributed according to a *finite mixture distribution* (see e.g., [8]) with $K$ components given by

$$p(x|\theta : K) = \sum_{i=1}^{K} \pi_i p(x|\theta_i), \tag{1}$$

where $\pi_i > 0$ $(i = 1, \cdots, K)$ and $\sum_{i=1}^{K} \pi_i = 1$. We set $\theta = (\pi_1, \cdots, \pi_{K-1}, \theta_1, \cdots, \theta_K)$. Here $\pi_i$ denotes the degree to what the $i$-th topic is likely to appear in a text stream. Note that each component in the mixture defines a single cluster in the sense of soft-clustering.

Throughout this paper we suppose that each $p_i(x|\theta_i)$ takes a form of a Gaussian density: Letting $d$ be the dimension of

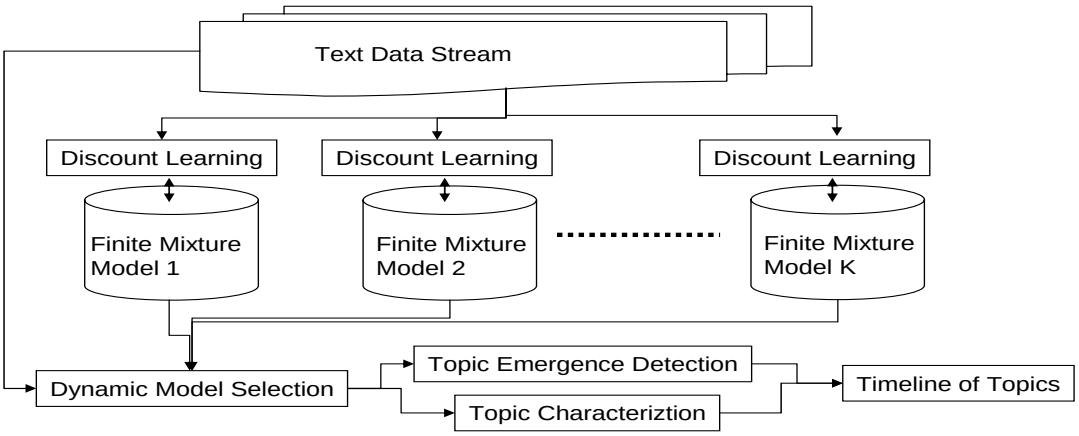


**Figure 1: Topic Trend Analysis System**

each datum,

$$p_i(x|\theta_i) = \phi_i(x|\mu_i, \Sigma_i) \quad (2)$$
$$= \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right),$$

where $\mu_i$ is a $d$-dimensional real-valued vector, $\Sigma_i$ is a $d \times d$-dimensional matrix, and we set $\theta_i = (\mu_i, \Sigma_i)$. In this case (1) is so-called a *Gaussian mixture*. Note that a Gaussian density may be replaced with any other form of probability distributions, such as a multinomial distribution.

In terms of a finite mixture model, a topic structure is identified by A) the number of components $K$ (how many topics exist), B) the weight vector $(\pi_1, \cdots, \pi_K)$ indicating how likely each topic appears, and C) the parameter values $\theta_i$ $(i = 1, \cdots, K)$ indicating how each topic is distributed. A topic structure in a text stream must be learned in an on-line fashion. Topic emergence detection is conducted by tracking the change of main components in the mixture model. Topic characterization is conducted by classifying each text into the component for which the posterior is largest and then by extracting feature terms characterizing the classified texts. Topic drift may be detected by tracking changes of a parameter value $\theta_i$ for each topic $i$. These tasks will be described in details in the sessions to follow.

The overall flow of the tasks is illustrated in Figure 1. A text is sequentially input to the system. We prepare a number of finite mixture models, for each of which we learn statistical parameters using the time-stamp based learning algorithm to perform topic identification. These tasks are performed in parallel. On the basis of the input data and learned models, we conduct dynamic model selection for choosing the optimal finite mixture model. We then compare the new optimal model with the last one to conduct topic emergence detection. Finally for each component of the optimal model, we conduct topic characterization.

## 3. TOPIC STRUCTURE IDENTIFICATION WITH DISCOUNTING LEARNING

In this section we propose an algorithm for learning a topic structure, which we call an *time-stamp based discounting topic learning algorithm.*

The algorithm is basically designed as a variant of the incremental EM algorithm for learning a finite mixture model (see, e.g., Neal and Hinton [9]). Our proposed one is distinguished from existing ones with regards to the following three main features:

1) *Adaptive to the change of the topic structure.* The parameters are updated by forgetting out-of-date statistics as time goes on. This is realized by putting a larger weight to the statistics for a more recent data.

2) *Making use of time stamps for texts.* Not only the time order of texts but also their time stamps are utilized to make the topic structure depend on the timeline. For example, for two text data $x_{t_1}, x_{t_2}$ $(t_1 < t_2)$, if the length $t_2 - t_1$ is larger, the topic structure learned at time $t_2$ will be less affected by that at time $t_1$.

3) *Normalizing data of different dimensions.* We consider the on-line situation where the dimension of a datum may increase as time goes by. This situation actually occurs because new words may possibly be added to the list of words every time a new text is input. Hence it is needed for normalizing data of different dimensions.

We suppose that text data $x_1, x_2, ...$ are given in this order, and each has a time-stamp indicating when it appeared. Here is a description of the algorithm, in which $\lambda$ is a discounting parameter, $\gamma_i$ denotes the posterior density of the $i$th component, and $m$ is introduced for calculation of weights for old statistics.

**Time-stamp Based Discounting Learning Algorithm**

**Initialization:**
Set initial values of $\pi_i^{(0)}, \mu_i^{(0)}, \Sigma_i^{(0)}, m^{(0)}$ $(i = 1, \cdots, k)$. Let $\alpha > 0,\ 0 < \lambda < 1$ be given.

**Iteration:**
For $t = 1, 2, ..$ do the following procedure.
For the $t$-th data be $x_t$ and its time stamp be $t_{new}$. Let the time stamp of the $(t-1)$-th data be $t_{old}$.
For $i = 1, \cdots, k$, update the parameters according to the

following rules:

$$\begin{aligned}
p(i|x_t) &:= \frac{\pi_i^{(t-1)} p_i(x_t|\mu_i^{(t-1)}, \Sigma_i^{(t-1)})}{\sum_{l=1}^{k} \pi_l^{(t-1)} p_l(x_t|\mu_l^{(t-1)}, \Sigma_l^{(t-1)})} \\
\gamma_i^{(t)} &:= \mathcal{WA}(p(i|x_t), 1/k|1, \alpha) \\
\pi_i^{(t)} &:= \mathcal{WA}(\pi_i^{(t)}, \gamma_i^{(t)}|m^{(t-1)}, \lambda^{-(t_{new}-t_{old})}) \\
\mu_i^{(t)} &:= \mathcal{WA}(\mu_i^{(t-1)}, x_t|\pi_i m^{(t-1)}, \lambda^{-(t_{new}-t_{old})}\gamma_i^{(t)}) \\
\Lambda_i^{(t)} &:= \mathcal{WA}(\Lambda_i^{(t-1)}, x_i x_i^T|\pi_i m^{(t-1)}, \lambda^{-(t_{new}-t_{old})}\gamma_i^{(t)}) \\
\Sigma_i^{(t)} &:= \Lambda_i^{(t)} - \mu_i \mu_i^T \\
m^{(t)} &:= \lambda^{(t_{new}-t_{old})} m^{(t-1)} + 1,
\end{aligned}$$

where $\mathcal{WA}$ denotes the operation such that

$$\mathcal{WA}(X, Y|A, B) = \frac{A}{A+B}X + \frac{B}{A+B}Y.$$

Generally, we set the initial value $\pi_i^{(0)} = 1/K, m^{(0)} = 0$, a small value to $\Sigma_i^{(0)}$, and set $\mu_i^{(0)}$ the first $x_t$s that are different each other. This algorithm updates $\pi_i$, $\mu_i$, and $\Sigma_i$ as the weighted average of the latest parameter value and the new statistics. The weight ratio is $m^{(t-1)} : \lambda^{-(t_{new}-t_{old})}$ for $\pi_i$, and $\pi_i m^{(t-1)} : \lambda^{-(t_{new}-t_{old})}\gamma_i^{(t)}$ for $\mu_i$ and $\Sigma_i$, respectively.

Note that Yamanishi et.al.'s sequentially discounting learning algorithm [14] can be thought of as a special case of this algorithm in which the time interval $t_{l+1} - t_l$ is independent of $l$. In that case if we further let $\lambda = 1$, the algorithm becomes an ordinary incremental EM algorithm.

In real implementation, we supposed that $\Sigma_i$ is a diagonal matrix for the sake of computational complexity issues. The scalability issue for dealing with a general matrix $\Sigma_i$ remains for future study.

# 4. TOPIC EMERGENCE DETECTION WITH DYNAMIC MODEL SELECTION

In this section we are concerned with the issue of *topic emergence detection*, i.e., tracking the emergence of a new topic. We reduce here this issue to that of selecting the optimal components in the mixture model dynamically. We call this statistical issue *dynamic model selection* (see also [7]).

The key idea of dynamic model selection is to first learn a finite mixture model with a relatively large number of components, then to select main components dynamically from among them on the basis of Rissanen's *predictive stochastic complexity* [10].

The procedure of dynamic model selection is described as follows:

**Initialization:**
Let $K_{max}$ (maximum number of mixture components) and $W$ (window size) be given positive integers.
Set initial values of $\pi_i^{(0)}, \theta_i^{(0)} = (\mu_i^{(0)}, \Sigma_i^{(0)})$ $(i = 1, \cdots, K_{max})$.

**Iteration:**
For $t = 1, 2, \cdots$, do the following procedure 1 to 4:

**1. Model Class Construction:**
Let $G_i^t$ be the window average of the posterior probability $(\gamma_i^{t-W} + \cdots + \gamma_i^t)/W$. For $k = 1, \cdots, K_{max}$, do the following procedure:

Let $\ell_1, \cdots, \ell_k$ be the indices of $k$ highest scores such that $G_{\ell_1}^{(t-1)} \geq \cdots \geq G_{\ell_k}^{(t-1)}$. Construct the following mixture model with $k$ components: For $s = t - W, \cdots, t$,

$$\begin{aligned}
p^{(t-1)}(x|\ell_1, \cdots, \ell_k) &= \sum_{j=1}^{k-1} \pi_{\ell_j}^{(t-1)} p_{\ell_j}(x|\theta_{\ell_j}^{(t-1)}) \\
&\quad + \left(1 - \sum_{j=1}^{k-1} \pi_{\ell_j}^{(t-1)}\right) \mathcal{U}.
\end{aligned}$$

where $\mathcal{U}$ is a uniform distribution over the domain.

**2. Predictive Stochastic Complexity Calculation:**
When the $t$-th input data $x_t$ with dimension $d_t$ is given, compute

$$S^{(t)}(k) = \sum_{s=t-W}^{t} \left(-\log p^{(s)}(x_s|\ell_1, \cdots, \ell_k)/d_s\right). \qquad (3)$$

**3. Model Selection:**
Select $k_t^*$ minimizing $S^{(t)}(k)$. Let $p_{\ell_j}(x|\theta_{\ell_j}^{(t-1)})$ $(j = 1, \cdots, k_t^*)$ be main components at time $t$, which we write as $\{C_1^{(t)}, \cdots C_{k_t^*}^{(t)}\}$.

**4. Estimation of Parameters:**
Learn a finite mixture model with $K_{max}$ components using the time-stamp based discounting learning algorithm. Let the estimated parameter be $(\pi_1^{(t)}, \cdots, \pi_{K_{max}}^{(t)}, \theta_1^{(t)}, \cdots, \theta_{K_{max}}^{(t)})$.

Note that the $S^{(t)}(k)$ can be thought of as a variant of Rissanen's *predictive stochastic complexity* [10] normalized by the dimension for each datum, which can be interpreted as the total code length required for encoding a data stream $x_{t-W}, ..., x_t$ into a binary string sequentially.

Once main components $C_1^{(t)}, \cdots, C_{k_t^*}^{(t)}$ are obtained, we compare them with $C_1^{(t-1)}, \cdots, C_{k_{t-1}^*}^{(t-1)}$ to check the emergence of a new topic or the disappearance of an existing topic in the following way. If a new component is selected at some point and remains for a longer time than a specified threshold, we may determine that a new topic has emerged. Specifically, if the optimal number $k_t^*$ of components becomes larger than $k_{t-1}^*$, we can recognize that a new topic has emerged. Similarly, if an existing component is not selected at some time and does not appear any longer, then we may determine that the topic has disappeared.

# 5. TOPIC CHARACTERIZATION WITH INFORMATION GAIN

Once the optimal finite mixture model is obtained, we are concerned with the issue of how to characterize each topic. We address this issue by extracting terms characterizing each topic and by observing the growth or decay of each topic component. Details are shown below.

A) *Extracting terms characterizing each topic.* We attempt to characterize each topic by extracting characteristic words for it. We perform this task by computing the information gain of possible words.

In the time-stamp based discounting topic learning algorithm, the posterior probability distribution over the set of clusters is estimated every time a text data is input. According to that posterior distribution an input text will be

categorized into the component for which the posterior probability is largest. This clustering task can be performed in an on-line fashion.

After observing the $t$-th datum $x_t$, for $i = 1, \cdots, k$, let $\mathcal{S}_t(i)$ be the set of texts in $x^t = x_1, .., x_t$ classified into the $i$-th component and let $t_i$ be the size of $\mathcal{S}_t(i)$. Let $\mathcal{S}_t = \cup_{i=1}^{k} \mathcal{S}_t(i)$.

Below we show the method for computing the information gain of a term $w$ for each topic component. For any term $w$, let $\mathcal{S}(w)$ be a set of vectors in $\mathcal{S}_t$ such that the frequency of $w$ is larger than a given threshold, and let $m_w$ be the size of $\mathcal{S}(w)$. Let $\mathcal{S}(\bar{w})$ be a set of vectors in $\mathcal{S}_t$ such that the frequency of $w$ is not larger than the threshold, and $m_{\bar{w}}$ be the size of $\mathcal{S}(\bar{w})$.

For a specified topic component, say, the $i$-th component, let $m_w^+$ be the number of vectors in $\mathcal{S}(w)$ that are also included in $\mathcal{S}_t(i)$. Let $m_{\bar{w}}^+$ be the number of vectors in $\mathcal{S}(\bar{w})$ that are also included in $\mathcal{S}_t(i)$.

Then we define the *information gain* of $w$ for the $i$-th topic component as follows:

$$IG(w|i) = I(t, t_i) - \left(I(m_w, m_w^+) + I(m_{\bar{w}}, m_{\bar{w}}^+)\right),$$

where $I(x, y)$ is the information measure such as stochastic complexity [10], extended stochastic complexity [13][5]. The *stochastic complexity* [10] is given as follows:

$$I(x, y) = xH\left(\frac{y}{x}\right) + \frac{1}{2}\log\left(\frac{x\pi}{2}\right),$$

where $H(x) = -x\log x - (1-x)\log(1-x)\log(1-x)$ is the binary entropy function, and log's base is 2. A special case of *extended stochastic complexity* is given as follows [13][5]:

$$I(x, y) = \min\{y, x - y\} + c\sqrt{x \log x},$$

where $c$ is a constant.

We select a specified number of terms $w$s of largest information gains. We can think of them the set of terms characterizing the $i$-th topic component.

The statistics: $m_w, m_w^+, m_{\bar{w}}, m_{\bar{w}}^+$ needed for computing information gain can be calculated in an on-line fashion. Hence topic characterization task is conducted in real-time.

B) *Observing the growth or decay of each cluster.* Let $G_i^{(t)}$ be the window average of the posterior probability of the $i$-th topic component, that is, $G_i^t = (\gamma_i^{t-W} +, \cdots, +\gamma_i^t)/W$. $G_i^{(t)}$ increases when texts corresponding to the $i$-th topic is input, and decreases when the other is input. We can see how rapidly this topic grows by observing $G_i^{(t)}$ as $t$ goes by.

## 6. EXPERIMENTAL RESULTS

We conducted an experiment on real data: contact data of a help desk for an internal e-mail service. An example of the data record is presented in Table 1. It has the field of contact date/time, question/request, answered date/time, answer, and so on. The number of the records is 1202. The date of the first/last contact is Feb 21 2004/May 20 respectively.

We input contact dates as the time-stamps, and questions/requests as the text data to our system. We set $K_{max}$ to 50 and $\lambda$ to 0.99. Our system ran on an NEC Express5800 with 1GHz Pentium III and a 1GB memory. The system was implemented using C, and OS was Windows 2000 Server. Processing 1202 records of data took about five minute.

Figure 2 shows the number of components $k_t^*$ selected by our system as main topics . The number increases at the beginning of March, and has a peak in the middle of April. Since a fiscal year begins at April in Japan, we can suppose that the number of topics at the help desk is increasing around the first day of April.

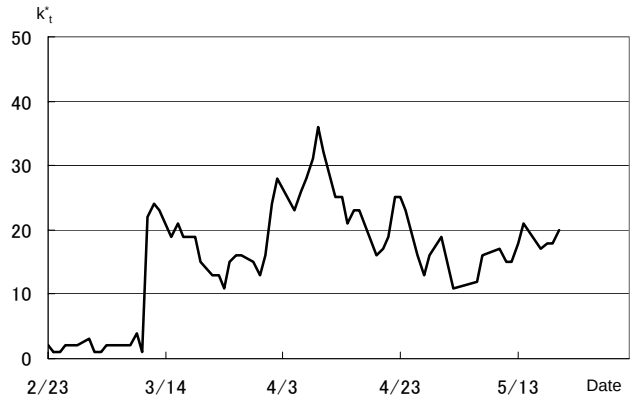


**Figure 2: Number of main topics**

Let us look into a few of the components, because we do not have enough space for all of the components. Here, we observe Component 27 and 42 in detail. Figure 3 shows the window averages $G_{27}, G_{42}$ of the posterior probabilities and the periods where the components are selected as main topics. $G_{27}$ increases in the beginning of April and has the first peak at April 12. Then it repeats increase and decrease until the middle of May. The corresponding component is selected as main from the first week of April, and remains as main until the middle of May (with short discontinuances). $G_{42}$ is positive during April, and also the corresponding topic is main during April. The lines of $G_i$s indicate how important the corresponding topics are in each time. Moreover, we can observe how the emerged topics grows and disappears from the figure. The topic corresponding to Component 42 emerges at the beginning of April, grows for two weeks, is attenuated, then drops out from the main topics at the end of April.

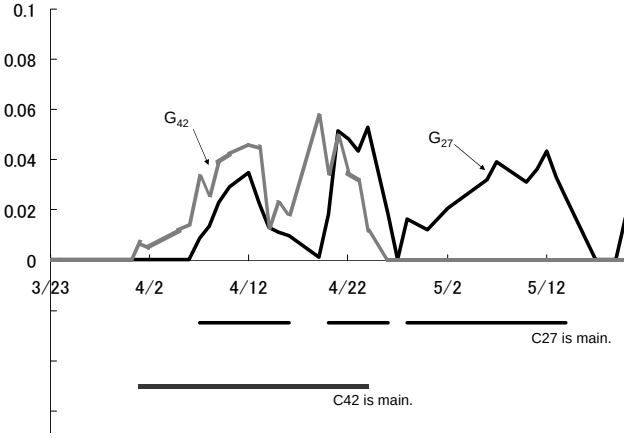


**Figure 3: $G_i$ and Period of Component 27, 42**

Term "transfer" was extracted as a characteristic word

**Table 1: Examples of help desk data records**

| Contact date/time | Question/Request | Answered date/time | Answer | ,,, |
|---|---|---|---|---|
| Feb 26 2004 14:05 | I forgot my password. How can I ... | Feb 26 2004 14:48 | You can get a new ... | |
| Feb 26 2004 14:08 | Until what time is an account for a ... | Feb 26 2004 14:25 | It is valid for 14 days after retirement. | |
| Feb 26 2004 14:09 | Is it possible to forward mails from ... | Feb 26 2004 15:09 | Yes. You can set up by .... | |
| .... | .... | .... | .... | |

for Component 27. Texts classified into this component are questions like "Is it possible to use Service XXX after I am transfered to YYY?". That kind of questions may increase around the beginning of a fiscal year. "Service ZZZ" and "failure" were extracted as chracteristic words for Component 42. Actually, Service ZZZ failed in the beginning of April, then, the topic consists of related complaints and questions.

In this way we can recognize the emergence, growth, and decay of each topic from the system. Through this example it has turned out that our framework for topic trend analysis are very effective for tracking dynamics of topic trends in contact data at a help desk.

## 7. CONCLUSION AND FUTURE STUDY

In this paper we have proposed a framework for tracking dynamics of topic trends using a finite mixture model. In this framework the three main tasks: topic structure identification, topic emergence detection, and topic characterization are unified within a single framework. Topic structure identification has been realized by our unique time-stamp based learning algorithm. It enables tracking topic structures adaptively by forgetting out-of-date statistics. Topic emergence detection has been realized on the basis of the theory of dynamic model selection. It enables detecting changes of the optimal number of components in the finite mixture model to check whether a new topic has appeared or not. Topic characterization has been realized by on-line text clustering and feature extraction based on information gain. Through the experiments using real data collected at a help desk, it is demonstrated that our framework works well in the sense that dynamics of topic trends can be tracked in a timely fashion.

The following issues remain open for future study:

*Context-based topic trend analysis:* In this paper we have proposed an approach to word-based topic trend analysis. However, we need to further analyze *contexts*, i.e., relations among words, in order to more deeply analyze the semantics of topics.

*Multi-topics analysis:* We supposed that one text comes from a single mixture component corresponding to a single topic. It is our future study how to deal with texts having multi topics.

## 8. REFERENCES


[1] J.Allen, R.Papka, and V.Lavrenko: On-line new event detection and tracking, in *Proceedings of SIGIR International Conference on Information Retrieval*, pp:37-45, 1998.

[2] X.Liu, Y.Gong, W.Xu, and S.Zhu: Document clustering with cluster refinement and model selection capabilities, in *Proceedings of SIGIR International Conference on Information Retrieval*, pp:191-198, 2002.

[3] S.Harve, B.Hetzler, and L.Norwell: ThemeRiver: Visualizing theme changes over time, in *Proceesings of IEEE Symposium on Information Visualization*, pp:115-123, 2000.

[4] J.Kleiberg: Bursty and hierarchical structure in streams, in *Proceedings of KDD2002*, pp:91-101, ACM Press, 2003.

[5] H.Li and K.Yamanishi: Text classification using ESC-based decision lists, *Information Processing and Management*, vol.38/3, pp:343-361, 2002.

[6] H.Li and K.Yamanishi: Topic analysis using a finite mixture model, Information Processing and Management, Vol.39/4, pp 521-541, 2003.

[7] Y.Matsunaga and K.Yamanishi: An information-theoretic approach to detecting anomalous behaviors, in *Information Technology Letters vol.2* (*Proc. of the 2nd Forum on Information Technologies*), pp:123-124, (in Japanese) 2003.

[8] G.McLahlan and D.Peel: *Finite Mixture Models*, Wiley Series in Probability and Statistics, John Wiley and Sons, 2000.

[9] R.M.Neal and G.E.Hinton: A view of the EM algorithm that justifies incremental sparse, and other variants, *Learning in Graphical Models*, M.Jordan (editor), MIT Press, Cambridge MA, USA.

[10] J.Rissanen: Universal coding, information, and estimation, *IEEE Trans. on Inform. Theory,* 30:629-636, 1984.

[11] R.Swan and J.Allen: Extracting significant time-varying features from text, in *Proceedings of 8th International Conference on Information Knowledge Management*, pp:38-45, 1999.

[12] R.Swan and J.Allen: Automatic generation of overview timelines, in *Proceedings of SIGIR International Conference on Information Retrieval*, pp:49-56, 2000.

[13] K.Yamanishi: A Decision-theoretic Extension of Stochastic Complexity and Its Applications to Learning, *IEEE Trans. on Inform. Theory,* vol.44/4, pp:1424-1439, 1998.

[14] K.Yamanishi, J.Takeuchi, G.Williams, and P.Milne: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," in *Proceedings of KDD2000*, ACM Press, pp:320–324 2000.

[15] Y.Yang, T.Pierce, J.G.Carbonell: A study on retrospective and on-line event detection, in *Proceedings of SIGIR International Conference on Information Retrieval*, pp:28-30, 1998.

[16] Y.Yang, J.Zang, J.Carbonell, and C.Jin: Topic-conditioned novelty detection, in *Proceedings of KDD 2002*, pp:688-693, 2002.