

Constructing Probabilistic Process Models based on Hidden Markov Models for Resource Allocation

Berny Carrera and Jae-Yoon Jung[†]

Dept. of Industrial and Management Systems Engineering, Kyung Hee University
1 Seocheon-dong, Giheung-gu, Yongin, Gyeonggi, Republic of Korea
{berny, jyjung}@khu.ac.kr

Abstract. A Hidden Markov Model (HMM) is a temporal statistical model which is widely utilized for various applications such as gene prediction, speech recognition and localization prediction. HMM represents the state of the process in a discrete variable, where the values are the possible observations of the world. For the purpose of process mining for resource allocation, HMM can be applied to discover a probabilistic workflow model from activities and identify the observations based on the resources utilized by each activity. In this paper, we introduce a process discovery method that combines an organizational perspective with a probabilistic approach to address the resource allocation and improve the productivity of resource management, maximizing the likelihood of the model using the Expectation-Maximization procedure.

Keywords: Probabilistic Process Discovery, Process Mining, Resource Allocation, Hidden Markov Models.

1 Introduction

The basic idea of business process modeling is how enterprises can represent actual processes in the way that those processes can be analyzed and improved. Through this model, organizations can obtain a graphical representation of the process activities defining a workflow in order to reach and accomplish intended objectives of business processes [1].

Currently, existing process models do not consider a probabilistic analysis of resources. Consequently, this paper examines a probabilistic approach to resource allocation to represent the relationship between resources and their activities as well as analyze their transitions to indicate the importance of each resource in the related activity. Hence, based on the context of process mining and their perspectives [2], this study will emphasize a control-flow perspective combined with an organizational perspective to support resource allocation by constructing a stochastic model from an event log and reveal how the resources interact directly with the activities.

[†] Corresponding author.

Meanwhile, machine learning and process mining techniques are closely related [3]. Some machine learning algorithms operate as a “black box” but are helpful for creating systems that can learn from data through poor control-flow discovery. Some machine learning techniques [4] are statistically based on the way that they have learned from data and are concerned with the collection, analysis, and interpretation of data, making these techniques the heart of quantitative reasoning which is necessary for making decisions and recommendations in systems. Hence, HMM is a technique that can deal precisely with noise, incompleteness and not over-fitting the model unlike other techniques e.g. decision trees, support vector machines, ANOVA, etc.

In general, HMM can be helpful in a specific form of the transition process and the process can be considered as a series of time slices. Basically, HMM is modeled after the process in different states for each activity, and the observations in HMM become the resources that interact with the activities. As a result, this research evaluates the possibility of discovering an extended probabilistic process model from a control-flow process with an organizational perspective by applying HMM for resource allocation, analyzing the resource attribute of each event and the activities flow, and estimating the parameters for each activity (state) and resource (observation).

Therefore, this paper investigates the contribution of the applicability of HMM for process mining based on previous premises by defining a probabilistic model that is likely to explain the observed behavior that considers the control-flow patterns, resources, and organization structure.

The remainder of the paper is organized as follows. First, the related work is discussed in more detail in Section 2. Section 3 introduces the event logs and HMM notations for the proposed model and explains HMM workflow discovery. Section 4 presents the estimation of the probabilities, resource allocation notations and applicability of the HMM Miner. Finally, Section 5 provides the conclusions and future work of the paper.

2 Related Work

There are a few research approaches of HMM in process mining. Rozinat et al. [5] propose a method that begins with a petri-net in order to construct a HMM for a quality evaluation of the model, taking into account different metrics like fitness and precision. Da Silva and Ferreira [6] apply HMM in sequence clustering, where all the elements of the sequence are related with each state in the model as well as propose five different topologies for the model. In [7] and [8], the intension mining approach is proposed to identify the reasoning of processes that are behind of the activities.

A number of previous studies have focused on resource and actions or reactions in the business process execution. Rozinat and van der Aalst [3] propose a decision mining technique based on case perspective analysis. Song and van der Aalst [10] present social mining that has an organizational perspective which monitors the originators and relationships and focuses on how the originators interact between them, though not directly with the activities. Recently, Kim et al. [11] introduced a performer recommendation using a decision tree which focuses on time and

organizational perspective, resulting in the length of the decision tree becoming too large and too difficult to analyze.

The proposed approach based on the probabilistic method differs from the existing studies from two perspectives: the control-flow perspective proposing a model based on the order of activities, and organizational perspective analyzing and proposing the best interaction between the activities and resources.

3 HMM-based Process Mining

3.1 Resource-Oriented Event Log

An event log is used as input of process mining to construct models that explain and interpret some aspect of the behavior stored. The event log contains traces referring to the number of cases, activities and resources, as shown in Table 1. It should be noted that this study has not taken into account time stamps or other types of data. Instead, the formal models presented by van der Aalst [12] are extended for HMM miner which adds a limitation based on an event log and a footprint matrix, and it is defined as a resource-oriented event log which follows:

Definition 1. (Resource-Oriented Event Log) Let L be an event log. L is a tuple $\langle C, S, V \rangle$, where C is the set of all possible cases, S of length N is the finite set of event labels $\{s_1, \dots, s_N\}$ that has been performed over L , and $V \in L^*$ of length M is a set of originators $\{v_1, \dots, v_M\}$ that specify the resource associated with the task.

The construction of HMM workflow is explained in Sections 3.2 and 3.3. To calculate the initial parameters, the event log is needed to mine the frequency of the occurrence of the activities and resources. This is explained more in Section 4. Section 5 discusses how to find the maximum-likelihood estimation (MLE).

Table 1. Example of process log.

Case ID	Activity
1	$a_{Pete}, b_{Sue}, c_{Sean}, d_{Mike}$
2	$a_{Ellen}, b_{Sean}, c_{Sue}, d_{Pete}$
3	$a_{Mike}, b_{Sue}, c_{Sue}, d_{Mike}$
4	$a_{Pete}, c_{Sean}, b_{Sue}, d_{Mike}$
5	$a_{Pete}, c_{Sean}, b_{Sue}, d_{Ellen}$
6	$a_{Ellen}, e_{Sue}, d_{Pete}$

3.2 HMM Miner Algorithm

To accommodate the event log, process mining, and HMM in terms of resource allocation, the concepts of HMM workflow is introduced. The formal notation for the HMM workflow is as follows:

Definition 2. (HMM Workflow) Let L be a resource-oriented event log. A HMM workflow $\theta(L) = (s, V, Fs, Fv)$ is represented by:

- $s = \{s_1, \dots, s_N, \varepsilon_{N+1}\} \in L$ where $\{s_1, \dots, s_N\}$ are the states or activities of L and $\{\varepsilon_{N+1}\}$ a dummy end state.
- $V = \{v_1, \dots, v_M, \varepsilon_{M+1}\} \in L$ where $\{v_1 \dots v_M\}$ are the observations of resources and $\{\varepsilon_{M+1}\}$ a dummy observation of the dummy end state $\{\varepsilon_{N+1}\}$.
- Fs is an $N \times N$ matrix with footprint sequence and frequency of occurrence for transitions s_i to s_j .
- Fv is an $N \times M$ matrix with the frequency of occurrence for resources V in state s_i .

The procedure for constructing the HMM workflow is presented in Fig. 1. The HMM miner algorithm first investigates the control-flow perspective of a process model from an event log, and considers the order of the events within a case. The attributes such as case id, activity, and resource are utilized in the mining process.

-
- 1: **Input:** event log L
 - 2: **Initialize**
 - 3: Find all activities S and all resources V in the traces in L .
 - 4: Find start activities S_s and end activities from S_e
 - 5: Create a footprint matrix with frequencies F_s from L
 - 6: Create a frequency matrix F_v from L
 - 7: Add the corresponding state s_i into the initial state set S_s with its starting probability π .
 - 8: Create dummy state $s_{N+1} = \varepsilon$ with an arc from ε to ε in S for refer the end of process
 - 9: Create dummy observation $v_{M+1} = \varepsilon$ to denote the end resource for state ε .
 - 10: All end activities given by S_e , the corresponding states s_i are end states; then create an arc from s_i to ε and calculate each frequency.
 - 11: **For** each pair (s_i, s_j) in S
 - 12: **If** $f_{s_{ij}}$ in Fs has a “direct follow” **then**
 - 13: Create an arc from s_i to s_j with its frequency ($f_{s_{ij}}$).
 - 14: **End For**
 - 15: **For** each pair (s_i, v_j) in F_v
 - 16: **If** $f_{v_{ij}} > 0$ **then**
 - 17: Create an arc from s_i to v_j with its frequency ($f_{v_{ij}}$).
 - 18: **End For**
 - 19: HMM workflow $\theta(L) = (s, V, Fs, Fv)$
 - 20: Calculate probabilities for A in arcs and B for related activities of v_i in each s_i .
 - 21: **Output:** initial HMM: $\lambda = (N, M, A, B, \pi)$
-

Fig. 1. HMM miner algorithm.

3.3 Mining Markovian States

HMM workflow learns from the event log which is fully observed data where the states of the HMM can be represented based on the activities. It takes the notation of event logs to describe the example log in Table 1 where it observes the event sequences $L = \langle abcd, abcd, abcd, acbd, acbd, aed \rangle$.

In order to discover a process model, the event log has to be analyzed while it is mining the causal dependencies by using the footprint notation for HMM Miner, and, when analyzed, different patterns of the states evaluate the precedence of each activity, and the frequency of the observed sequences. In order to satisfy these requirements the footprint matrix is introduced in Definition 3.

In process mining, α -algorithm [9] is considered one of the baseline techniques for analysis of patterns in event logs. In our approach, a footprint matrix is required to construct the state transitions of a HMM model analyzing the relations of a pair of activities. The footprint matrix is focused on a process perspective especially in analysis and differentiation of activities sequences. It can measure the frequency of activities and the sequences, helping to real situations where the data is not complete or has noise.

Definition 3. (Footprint Matrix for HMM Miner) Let L be an event log. Let $x_1, x_2 \in L$:

- $x_1 >_L x_2$ if and only if there is a trace $\sigma = \langle t_1, t_2, t_3, \dots, t_n \rangle$ and $i \in \{1, \dots, n-1\}$ such that $\sigma \in L$ and $t_i = x_1$ and $t_{i+1} = x_2$ (Contains all pairs of activities in a “directly follows”). If is a directly follow, $fs = \sum_{\sigma \in L} L(\sigma)$
- $x_1 \rightarrow_L x_2$ if and only if $x_1 >_L x_2$ and $x_2 \not>_L x_1$ (Contains all pairs of activities in a “causality” relation)
- $x_1 \#_L x_2$ if and only if $x_1 \not>_L x_2$ and $x_2 \not>_L x_1$
- $x_1 \parallel_L x_2$ if and only if $x_1 >_L x_2$ and $x_2 >_L x_1$

Table 2. Footprint matrix F_s for event log in Table 1.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>a</i>	#	$\rightarrow, 3$	$\rightarrow, 2$	#	$\rightarrow, 1$
<i>b</i>	\leftarrow	#	$\parallel, 3$	$\rightarrow, 2$	#
<i>c</i>	\leftarrow	$\parallel, 2$	#	$\rightarrow, 3$	#
<i>d</i>	#	\leftarrow	\leftarrow	#	\leftarrow
<i>e</i>	\leftarrow	#	#	$\rightarrow, 1$	#

For our implementation activities $x_1, x_2 \dots x_n$, are the states for the HMM, e.g. *a*, *b*, ..., *e*. After constructing the footprint matrix, the next step is to start with the construction of a Markov chain that simulates a dependency graph based on the states of HMM. The HMM diagram has states expressed by “oval shape and activity name,” and transitions are represented by two different “arcs,” one for transition activities and other for the resource relationships. The first one represents the successor states and in terms of resource mining, the relationship between the activities, and the second,

the observations (resources) that correspond to that activity, e.g. activity a has a causality relation with activities b , c and e ; so, it creates a transition denoted by an “arc” from a to these activities. The initial states of the HMM workflow are denoted graphically by a double oval, and the probabilities of each initial state are assigned in the initial state distribution π . Fig. 2 shows that all observations were represented with a “square shape and resource name” and assigned directly to each state, and, in turn, the resources are mined from the event log and related to specific activity.

To model the end of the process dummy state ϵ is added, which receives its input from the last activity in each sequence and from which no other state is reachable. Consequently, the transition probability from this state to itself should equal 1. Figure 2 describes a constructed HMM workflow and the interaction of the activities with the different resources show how after the state d is fired and reaches the state ϵ .

The next section describes how to estimate the probabilities in the mode which first, calculates the initial values for the state transition probability distribution A and for the observation symbol probability distribution B . After several steps, it is determined that the expectation maximization for these probabilities obtains the estimated A^* and B^* .

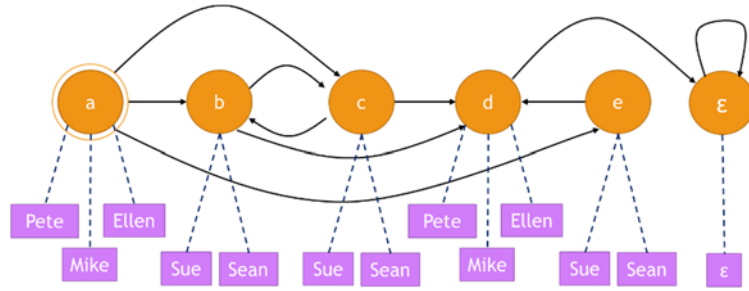


Fig. 2. HMM workflow constructed from the footprint matrix without probabilities of Table 2.

4 Resource Allocation with HMM Miner

4.1 Relating Resources to HMM

After obtaining the mined structure of HMM, including the number of states (activities) N , name of the different states s , count of observations (resources) M , label of each observation V and the initial states defined in π , the next step is to determine the initial state transition probability distribution A and the observation symbol probability distribution B . To begin analyzing both probability distributions, the frequency f of each activity and resource, is needed and with them, the weight of the arcs can be calculated, and it is possible to decide which elements should be kept for analysis, and which elements can be discarded to reduce the noise of the mined event log. Starting with the transition probabilities, the footprint matrix is needed to obtain the frequencies.

In order to start the analysis to get the probabilities for the HMM workflow the assumption is made that the data is from an event log $L = \langle C, S, V \rangle$; and each event in L is given by a tuple (s, v) and the total events in L is K . Then $(s_k, v_k) = (s_1, v_1), \dots, (s_N, v_M)$ the probabilities can be computed and collected :

$$\prod_k P(s_k, v_k) . \quad (1)$$

$$P(s, v) = \prod_i \pi_i^{f(i)} \cdot \prod_{i,j} a_{i,j}^{f(i,j)} \cdot \prod_{i,o} b_i(v_k)^{f(v_k)} . \quad (2)$$

Note in this equation, it is taken in account the frequency where $f(i)$ is the number of times i is the initial state in (s, v) . In $f(i, j)$ the number of times j follows i in (s, v) this frequency is in the footprint matrix Fs and $f(v_k)$ the number of times i is paired with observation v is in matrix Fv .

According to these HMM parameters, the log probability can be easily calculated:

$$L(\lambda) = \sum_{l=1}^k \log P(s_l, v_l) . \quad (3)$$

$$\begin{aligned} \text{Log}(\lambda) = & \sum_{l=1}^k \sum_i f(i) \log(\pi_i) + \sum_{i,j} f(i, j) \log(a_{i,j}) \\ & + \sum_{i,v_k} f(v_k) \log(b_i(v_k)) . \end{aligned} \quad (4)$$

In order to find λ that give us a maximum value for $L(\lambda)$ The following formula is needed: $d L(\lambda)/d\lambda = 0$

$$\pi_i = \frac{\sum_i f(i)}{\sum_i \sum_h f(h)} . \quad (5)$$

$$a_{i,j} = \frac{\sum_i f(i, j)}{\sum_i \sum_h f(i, h)} . \quad (6)$$

$$b_i(v_k) = \frac{\sum_i f(i)}{\sum_i \sum_{v_k \in V} f(i, v_k)} . \quad (7)$$

Basically, a supervised learning from all mined activities and resources is taken, allowing for the calculation of the probability of each element independently.

4.2 Maximum Likelihood Estimation

Knowing the initial values $\lambda = (N, M, A, B, \pi)$; it would be possible to estimate the maximum likelihood in selecting the best values for the model parameters that make

the observed data the most probable and obtain $\lambda^* = (N, M, A^*, B^*, \pi^*)$. As described in Fig. 3, EM procedure essentially calculates their values in 2 steps E and M can be calculated using forward and backward algorithms [4] for HMM.

1:	Inputs: $\lambda = (N, M, A, B, \pi)$ and event log L
2:	Initialize
3:	Repeat
4:	Using forward algorithm calculate α_i for each trace in L
5:	Using backward algorithm calculate β_i for each trace in L
6:	Re-estimation of temporal variables
7:	Calculate γ and ξ based on $\sum \alpha_i$ and $\sum \beta_i$
8:	Calculate A^*, B^*, π^* from the temporal variables
9:	Updating λ
10:	Until A and B do not change
11:	Output: $\lambda^* = (N, M, A^*, B^*, \pi^*)$

Fig. 3. Expectation-Maximization procedure.

In order to improve λ , the EM procedure uses α_i from the forward algorithm and β_i from the backward algorithm, which utilizes the variables γ and ξ as temporal variables [4] in order to estimate the values of A^*, B^* , and π^* .

$$\gamma_i(t) = \frac{\alpha_i(t) * \beta_i(t)}{\sum_{j=1}^N \alpha_j(t) * \beta_j(t)}. \quad (8)$$

$$\xi_{ij}(t) = \frac{\alpha_i(t) * a_{ij} * \beta_j(t+1) * b_j(v_{t+1})}{\sum_{p=1}^N \sum_{q=1}^N \alpha_p(t) * a_{pq} * \beta_q(t+1) * b_q(v_{t+1})}. \quad (9)$$

Given an observation sequence, $\gamma_i(t)$ is the probability of being in state i at time t and $\xi_{ij}(t)$ is the probability of being in state i at time t and being in state j at time $t+1$. These temporal variables are useful in order to determine the maximum likelihood of the parameters, $\gamma_i(t)$ can be interpreted like the possibility to be in a specific activity on a determined time of the process and $\xi_{ij}(t)$ the probability of the following activities in a specified time.

$$\pi_i^* = \gamma_1. \quad (10)$$

where π_i^* is the expected initial state distribution at time 1 in state i . In other words, the expected probability is to begin the process in a specific activity.

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}. \quad (11)$$

where a_{ij}^* is the expected state transition probability distribution, and is the number of times that the transition i occurs in sequence from $t=1$ to $T-1$. Then, a_{ij}^* is the expected probability from being in activity i to j .

$$b_j^*(v_k) = \frac{\sum_{t=1}^{T-1} \gamma_j(t)}{\sum_{t=1}^{T-1} \gamma_j(t)}. \quad (12)$$

$b_j^*(v_k)$ is the expected observation symbol probability distribution, where is the expected number of times in state j and observing symbol v_k , divided by the expected number of times in j . In terms of this study, it is the predicted parameters of the resources assigned to a specified activity.

4.3 Example

The example is based on the event log from Table 1. It is used for the mining algorithm in Fig. 1 in order to construct the HMM workflow. In Table 2, a footprint matrix is created and Fig. 2 shows the mined HMM workflow without the frequencies. From this point, the next step is calculate the frequency matrix Fv for the analysis of resources, and calculates the initial parameters explained in Section 4.1, and the values for the maximum likelihood explained in Section 4.2.

First, the mined event log produces three different activity sequences and the algorithm estimate a percentage for $\langle abcde \rangle$ with 0.5%, $\langle acbde \rangle$ with 0.33% and $\langle aed\epsilon \rangle$ with 0.16%, the number of states that identifies each activity are $N = 6$ with $s = \{a, b, c, d, e, \epsilon\}$ and the number of resources that will be the observations $M = 6$ with $V = \{\text{Pete, Mike, Ellen, Sue, Sean, } \epsilon\}$.

Table 3. Frequency matrix Fv for event log in Table 1.

	Pete	Mike	Ellen	Sue	Sean	ϵ
a	3	1	2	0	0	0
b	0	0	0	4	1	0
c	0	0	0	2	3	0
d	2	3	1	0	0	0
e	0	0	0	1	0	0
ϵ	0	0	0	0	0	1

Table 3 shows the frequency of resources s_i based on activities v_j and is used to create the observation symbol probability distribution $B = \{b_i(v_k)\}$. The initial state distribution is $\pi_i = [1,0,0,0,0,0]^T$, where activity $\{a\}$ is the only one that starts all the process sequence.

Table 4. State transition probability distribution $A = \{a_{i,j}\}$

	a	b	c	d	e	ϵ
a	0	3/6	2/6	0	1/6	0
b	0	0	3/5	2/5	0	0
c	0	2/5	0	3/5	0	0
d	0	0	0	0	0	1
e	0	0	0	1.0	0	0
ϵ	0	0	0	0	0	1.0

Table 5. Observation symbol probability distribution $B = \{b_i(v_k)\}$

	Pete	Mike	Ellen	Sue	Sean	ϵ
a	3/6	1/6	2/6	0	0	0
b	0	0	0	4/5	1/5	0
c	0	0	0	2/5	3/5	0
d	2/6	3/6	1/6	0	0	0
e	0	0	0	1.0	0	0
ϵ	0	0	0	0	0	1.0

Table 4 shows the probabilities for HMM workflow from equation 6 where it analyzes the transition from the state i corresponding to the current transition to all states j . For example to assign value probability from activity $\{a\}$ to $\{b\}$, after $\{a\}$ is fired $\{b, c, e\}$ can be the next state, the total frequency of sequences $\{ab\}$, $\{ac\}$ and $\{ae\}$ are 6, and evaluating just $\{ab\}$ the frequency is 3 showed in Table 2, the probability is 3/6. In the case of resources probabilities, Table 5 shows the observation probability distribution where it is analyzes the resources that interact with a specific activity. For example, in activity $\{a\}$ the resources used are $\{Pete, Mike, Ellen\}$ the total frequency is 6 and frequency observed for $\{Mike\}$ is 1 showed in Table 3, then applying equation 7 the probability is 1/6.

Table 6. Expected state transition probability distribution $A^* = \{a_{ij}^*\}$

	a	b	c	d	e	ϵ
a	0	0.53298	0.3987	0	0.6832	0
b	0	0	0.51998	0.48002	0	0
c	0	0.42249	0	0.57751	0	0
d	0	0	0	0	0	1.0
e	0	0	0	1.0	0	0
ϵ	0	0	0	0	0	1.0

With the initial values for the HMM workflow, the EM procedure is utilized to estimate the expected values for transition A^* and observation B^* distribution. Tables 6 and 7 are shown in graphical form in Fig. 3.

Table 7. Expected observation symbol probability distribution $B^* = \{b_j^*(v_k)\}$

	Pete	Mike	Ellen	Sue	Sean	ϵ
a	0.5	0.16667	0.33333	0	0	0
b	0	0	0	0.79102	0.20898	0
c	0	0	0	0.44717	0.55283	0
d	0.33333	0.5	0.16667	0	0	0
e	0	0	0	1.0	0	0
ϵ	0	0	0	0	0	1.0

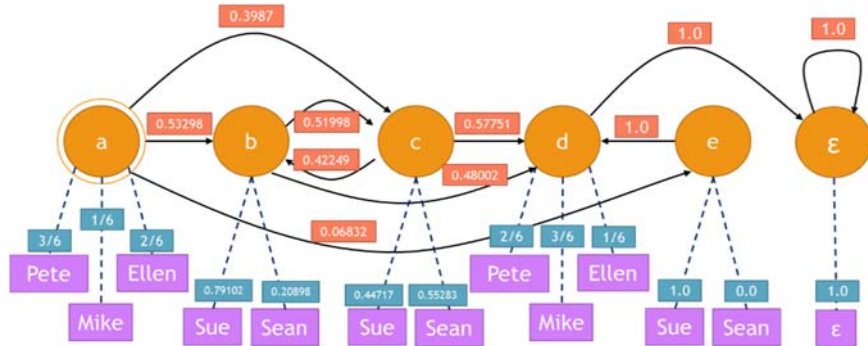


Fig. 4. Estimated HMM workflow constructed from the footprint matrix of Table 1.

It should be noted that this study has been primarily concerned with the estimation of control flow probabilities and the analysis of work distributions between resources and activities. Our approach is helpful to answer related questions in resource allocation, such as: “How to determine the experience of resources in a specific activity?,” “How to promote a resource effectively based on their experience?” and “What activity has to be scheduled and the resources required?.”

We plan to conduct related experiments in the near future in order to compare and evaluate the proposed method using real-life event logs and implement it in a ProM framework.

5 Conclusions

This study presented designs a probabilistic discovery process from event log using HMM to support resource allocation. Specifically, the expectation maximization approach was adopted for estimating the model parameters. The proposed method is useful in real-world scenarios for managing standard errors and noise in real case event logs. Since determining the number of hidden states is very difficult, the following model was based on activities and resources in such a way that the comprehension of the model is enhanced. Also, the proposed technique is helpful to compare the performance of resources for activity executions. Future work includes the consideration of the time perspective to analyze if a resource is busy and the extension of the technique to other scenarios.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2012R1A1B4003505).

References

1. Aguilar-Saven, R. S.: Business process modelling: Review and framework. *Int. J. Prod. Econ.*, 90(2), 129-149 (2004)
2. van der Aalst, W.M.P., Adriansyah, A., de Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Pontieri, L. et al.: Process mining manifesto. In: Florian D., Kamel B., Schahram D. (eds.) *BPM Workshop 2011*. LNCS, vol. 99, pp. 169-194. Springer Berlin, Heidelberg (2012)
3. Rozinat, A., & van der Aalst, W.M.P.: Decision mining in ProM. In: Dustdar, S., Fiadeiro, J.L., Sheth, A. (eds.) *BPM 2006*. LNCS, vol. 4102, pp. 420-425. Springer, Heidelberg (2006)
4. Bishop, C.M.: *Pattern recognition and machine learning*. Springer, New York (2006)
5. Rozinat, A., Veloso, M., & van der Aalst, W.M.P.: Using hidden markov models to evaluate the quality of discovered process models. *Extended Version*. BPM Center Report BPM-08-10, BPMcenter.org (2008)
6. da Silva, G.A., & Ferreira, D.R.: Applying hidden Markov models to process mining. In *Sistemas e Tecnologias de Informação*. AISTI/FEUP/UPF (2009)
7. Khodabandelou, G.: Supervised intentional process models discovery using hidden markov models. *RCIS 2013*, 1(11). IEEE Press (2013)
8. Khodabandelou, G., Hug, C., Deneckère, R., & Salinesi, C.: Process mining versus intention mining. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) *BPMDs 2013*. LNBIP, vol. 147, pp.466-480. Springer Berlin, Heidelberg (2013)
9. van der Aalst, W.M.P., Weijters, T., & Maruster, L.: Workflow mining: Discovering process models from event logs. *IEEE T. Knowl. Data En.*, 16(9). 1128-1142 (2004)
10. Song, M., & van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. *Decis. Support. Syst.*, 46(1), 300-317 (2008)
11. Kim, A., Obregon, J., & Jung, J.-Y.: Constructing decision trees from process logs for performer recommendation. *BPM Workshops 2013*. LNCS, vol. 171, pp. 224-236. Springer, Heidelberg (2014)
12. van der Aalst, W.M.P.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, Heidelberg (2011)