Abstract. The possibility of handling infrequent, higher density, additional loads, used mainly for on-line characterization of workloads, is considered. This is achieved through a sliding version of a hidden Markov model (SlidHMM). Essentially, a SlidHMM keeps track of processes that change with time and the constant size of the observation set helps reduce the space and time complexity of the Baum-Welch algorithm, which now need only deal with the new observations. Practically, an approximate Baum-Welch algorithm, which is incremental and partly based on the simple moving average technique, is obtained, where new data points are added to an input trace without re-calculating model parameters, whilst simultaneously discarding any outdated points. The success of this technique could cut processing times significantly, making HMMs more efficient and thence synthetic workloads computationally more cost effective. The performance of our SlidHMM is validated in terms of means and standard deviations of observations (e.g. numbers of operations of certain types) taken from the original and synthetic traces.

1 Introduction

The hidden Markov model (HMM) has been relatively popular in workload characterization [3] in recent years. Its parsimony, portability and efficient training, through its expectation maximization algorithm, has made it useful for reproducing representative workload traces for simulating live systems. Research has also complimented these applications through an incremental storage model [9,12], on which quantitative measures were made. This work has proven that computation time for a reliably parameterized model can be significantly reduced, whilst maintaining accuracy of the model. Indeed, the incremental approach, by which a model’s parameters are progressively updated rather than periodically re-calculated, has been appealing in terms of run-time performance.

1.1 Background

To achieve an incremental model, one can adapt the standard HMM algorithms used to train the model. These statistical algorithms under investigation are essentially those solving the three fundamental problems associated with HMMs:
firstly, obtain $P(O; \lambda)$, or the probability of the observed sequence $O$ given the model $\lambda$; secondly, maximize $P(O; \lambda)$ by adjusting the model parameters for a given observation sequence $O$; thirdly, determine the most likely hidden state sequence for an observed sequence. These three problems are solved by three respective algorithms: using the Forward-Backward algorithm \cite{1}, the Baum-Welch algorithm\cite[2] and the Viterbi algorithm \cite[10]. The solutions to the Forward-Backward and Baum-Welch algorithms are presented in the following sections.

1.2 Forward-Backward Algorithm

The Forward-Backward algorithm aims to find $P(O; \lambda)$, which is the probability of the given sequence of observations $O = (O_1, O_2, \ldots, O_T)$ given the model $\lambda = (A, B, \pi)$, where there are $T$ observations, $A$ is the state transition matrix, $B$ is the observation matrix and $\pi$ is the initial state distribution. This is equivalent to determining the likelihood of the observed sequence $O$ occurring. We use the same format presented in \cite[17], which is based partly on Rabiner’s solution \cite[13,14]. Initially, the focus is on the $\alpha$-pass, which is the “forward” part of the Forward-Backward algorithm. Then, we shift our attention to the corresponding $\beta$-pass, aka. the “backward” part of the algorithm.

To begin with, we define $\alpha_t(i)$ as the probability of obtaining the observation sequence up to time $t$ together with the state $q_i$ at time $t$, given our model $\lambda$. Using $N$ as the number of states and $T$ as the number of observations, the mathematical notation is

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, s_t = q_i; \lambda)$$  \hspace{1cm} (1)

where $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$, and $s_t$ is the state at time $t$.

Proceeding inductively, we write the solution for $\alpha_t(i)$ as follows:

1. For $i = 1, 2, \ldots, N$,

$$\alpha_1(i) = \pi_i b_i(O_1).$$

2. For $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T - 1$,

$$\alpha_{t+1}(i) = \sum_{j=1}^{N} \alpha_t(j) a_{ji} b_i(O_{t+1})$$

where $\alpha_t(j) a_{ji}$ is the probability of the joint event observing $O_1, O_2, \ldots, O_t$ and moving from state $q_j$ at time $t$ to state $q_i$ at time $t + 1$.

3. It follows that,

$$P(O; \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

where $\alpha_T(i) = P(O_1, O_2, \ldots, O_T, s_T = q_i; \lambda)$

\hspace{1cm} 1 This algorithm uses the Forward-Backward algorithm iteratively.