Patient-specific early classification of multivariate observations

Mohamed F. Ghalwash

Center for Data Analytics and Biomedical Informatics,
Temple University,
Philadelphia, PA, USA
and
Mathematics Department, Faculty of Science,
Ain Shams University,
Cairo, Egypt
Email: mohamed.ghalwash@temple.edu

Dušan Ramljak and Zoran Obradović*

Center for Data Analytics and Biomedical Informatics,
Temple University,
Philadelphia, PA, USA
Email: dusan.ramljak@temple.edu
Email: zoran.obradovic@temple.edu
*Corresponding author

Abstract: Early classification of time series has been receiving a lot of attention recently. In this paper we present a model, which we call the Early Classification Model (ECM), that allows for early, accurate and patient-specific classification of multivariate observations. ECM is comprised of an integration of the widely used Hidden Markov Model (HMM) and Support Vector Machine (SVM) models. It attained very promising results on the datasets we tested it on: in one set of experiments based on a published dataset of response to drug therapy in Multiple Sclerosis patients, ECM used only an average of 40% of a time series and was able to outperform some of the baseline models, which needed the full time series for classification. In the set of experiments tested on a sepsis therapy dataset, ECM was able to surpass the standard threshold-based method and the state-of-the-art method for early classification of multivariate time series.

Keywords: bioinformatics; patient-specific classification; early classification; multivariate time series; hybrid models; HMM; SVM; gene expression; sepsis therapy; early diagnosis.

Reference to this paper should be made as follows: Ghalwash, M.F., Ramljak, D. and Obradović, Z. (2015) 'Patient-specific early classification of multivariate observations', *Int. J. Data Mining and Bioinformatics*, Vol. 11, No. 4, pp.392–411.

Biographical notes: Mohamed F. Ghalwash is a PhD candidate of Computer and Information Science in Temple University, Philadelphia, USA. He obtained his MSc degree, titled 'Randomized Algorithms and their Applications', from Ain Shams University, Cairo, Egypt. His research interests are machine learning, data mining, bioinformatics and early classification.

Dušan Ramljak is a PhD candidate of Computer and Information Science in Temple University, Philadelphia, PA, USA. He obtained both his BSc and MSc degrees from School of Electrical Engineering, University of Belgrade, Belgrade, Serbia. His research interests are machine learning, data mining, bioinformatics and mining temporal networks.

Zoran Obradović, Temple University CIS Department Professor, secondary appointment in Statistics, and Director of the Center for data Analytics and Biomedical Informatics, is the Executive Editor at the *Journal on Statistical Analysis and Data Mining* – the official publication of the American Statistical Association, and is an editorial board member at 11 journals. He is general co-chair for 2013 and 2014 SIAM International Conference on Data Mining and was the program/track chair at many data mining, and biomedical informatics conferences. His research interests include data mining, machine learning and complex networks applications in climate modelling and health management.

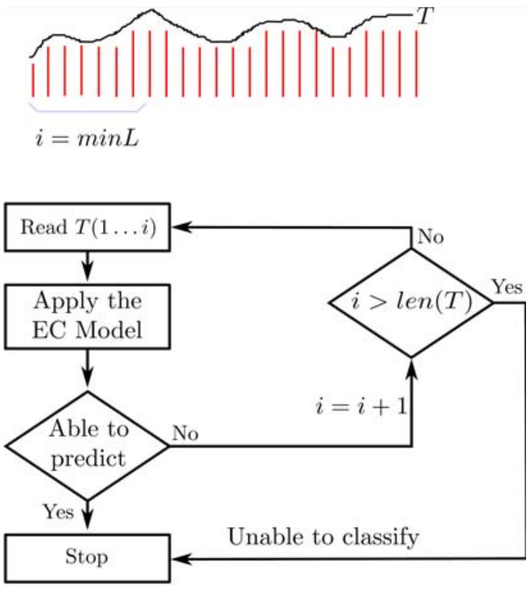
This paper is a revised and expanded version of a paper entitled 'Early classification of multivariate time series using a hybrid HMM/SVM model', presented at the 'IEEE International Conference on Bioinformatics and Biomedicine', Philadelphia, PA, USA, 4–7 October 2012.

1 Introduction

A time series of several classes can be created by observing various patient attributes over time. Such data can be used for experiments involving feature selection and clinically relevant temporal pattern identification, among other uses. Since many clinical applications tend to be very time sensitive, being able to use a shorter time interval for classification is often more favorable than having a slightly more accurate result. For example, observing only the first portion of a patient's data may allow doctors to identify a deadly condition before it has time to fully manifest or to be able to safely stop treatment after it has cured the patient and before it can do unnecessary damage. Early classification techniques are therefore beneficial for this realm, as they offer the very favorable tradeoff of using less of a time series at the cost of some accuracy of the result. The challenge of adapting early classification for biomedical purposes is ensuring that the result is obtained early enough to make a difference and also accurate enough for the doctor to be confident in the model's result. The goal of this paper is to introduce an Early Classification Model (ECM) that, by training on a full time series, offers classification at a very early time point during the testing phase, while staying competitive in terms of accuracy with other models that use full time series both in training and testing.

In an early classification context (see Figure 1), the objective is to provide patient specific classification of unknown time series as early as possible. Therefore, instead of utilising the whole time series, the early classification method looks into a portion (current stream) of the unknown time series and determines whether it is able to predict the label of the whole time series without looking at the rest of the time series. If the method is able to predict at the time point which is at the end of the current stream, the label is predicted. Otherwise, the method requires more data for the unknown time series and looks at a larger segment and does so until it is able to predict the label of the time series.

Figure 1 Early classification framework (see online version for colours)



Notes: Early Classification (EC) method looks into a portion *T*(1…*i*) of length *i* of the unknown time series *T*. The early classification Model is then applied to the portion *T*(1…*i*) and the process is repeated (adding new time points) until the label is predicted or the end of the time series is reached without obtaining a prediction.

Analysis of biomedical data has been a popular topic for many years, although using this kind of data for early classification is a far less explored field. Models that are naturally able to leverage temporal information, such as HMM, have been shown to be particularly effective for biomedical data since the temporal aspect plays a vital part in describing the data. Furthermore, since HMMs are resilient to having missing values in the data, they seem like a natural fit for biomedical applications, which often contain a significant number of missing values due to the various human and technological factors involved in obtaining the data. Lin et al. (2008) adapt HMMs to classify instances based on time series gene expression data and their results reinforce the effectiveness of HMMs in a biomedical context.

Another important issue often encountered when dealing with gene expression time series data is the varied response rates problem. Due to the differences in internal chemistries of each patient, the same condition can take different amounts of time to manifest in different individuals. Thus, while the overall pattern of the expression profile for a particular condition may be very similar in each patient, it can show up as stretched or compressed in time, making it much harder to detect accurately. By building a hybrid model we were able to not only provide early classification for multivariate time series, but also address this problem of detection.

Although attaining accurate classification is often the primary goal of machine learning task, in many applications it is important to attain decisions that are not only accurate, but can also be easily interpreted and obtained early. For example, diseases like sepsis, a systemic inflammatory response syndrome triggered by infection, are often diagnosed too late. The sepsis therapy problem is compounded by the fact that therapy is delayed until the patient is transported to an intensive care unit. Delay in treatment after sepsis is diagnosed is a serious problem, as it may have dire consequences for the patient. It is reported that for every seven hours that the administration of appropriate therapy is delayed, the mortality rate increases by about 7% (Dellinger et al., 2008). Detecting the patient's condition as early as possible can save his or her life. Early classification provides the opportunity to administer an appropriate therapy early enough for therapy to have a strong effect. As a result, early classification is highly desirable in this context.

In this paper, we introduce a hybrid model specifically designed for early and accurate classification of multivariate biomedical time series data. In addition to offering good performance and outperforming several baseline methods applied to a number of datasets, we successfully address the early classification requirement of sepsis therapy.

The remainder of the paper is organised as follows. Section 1.1 discusses some of the related work pertinent to this paper, including works discussing the potency of HMMs and their applications to gene expression data. In Section 2 we provide an explanation of our hybrid model as well as an in-depth description of how the hybrid model is trained and used for early classification. Section 3 includes a description of the various datasets we used, an explanation of our experimental setup and the results we obtained in our experiments. Finally, Section 4 includes concluding remarks and possible avenues for future research.

1.1 Related work

Due to their flexibility and popularity in the field of statistics, HMMs have long been the model of choice for a variety of applications, including speech recognition, natural language processing and, more recently, gene expression analysis. In his timeless tutorial on HMMs, Rabiner (1989) provided the first in-depth examination of the mathematics behind these models in less abstract environments, offering insights into how these models could be used to effectively model real world systems. His analysis of the application of HMMs for speech recognition spurred researchers to adapt HMMs for this purpose, generating a vast number of papers on the topic. In his survey, Jiang (2010) summarises a number of these papers, focusing on methods that utilise discriminative training techniques to provide significant improvements in the field of speech recognition. However, in our experiments it required more computational time, without an improvement in performance.

As the aforementioned survey shows, the notion of combining generative and discriminative techniques into a single model has been around for some time. One paper that offers a very intuitive and adaptable approach to creating such a model, which was not included in the survey, was written by Huang et al. (2006). Although Huang's model was designed for the task of online handwriting symbol recognition, we found that the underlying approach used to build this model could easily be adapted to the realm of early classification of gene expression time series. Despite the very different natures of the handwriting symbol and gene expression time series datasets, we were able to create an ECM using the same general approach. The method inherits the practical and dynamical modeling abilities from HMMs and its robust discriminating ability derived from Support Vector Machines (SVMs) is vital for classification tasks. However, we are exploring every relevant segment length at every possible starting point in a manner which will be explained in the next section and that was not done in previous work.

The application of HMMs to the field of gene expression time series classification is much more recent than their uses in speech recognition, but there has already been a significant amount of research performed in this area. Among the earliest efforts of utilising the potency of HMMs for gene expression analysis was Schliep et al. (2003) paper. Although focused on clustering the gene expression time series, rather than on classifying them, the authors nonetheless managed to show the utility of HMMs in the context of gene expression. This utility was attained primarily due to the fact that HMMs leverage the temporal dependencies of the data, whereas methods that ignore these dependencies lose a lot of information, as evidenced by the difference in the quality of the clusters generated by both kinds of methods.

The work of Lin et al. (2008) is also noteworthy, as they were one of the forerunners of using HMMs for classification of gene expression time series. However, despite the novelty and efficiency of Lin's model, it was incapable of performing early classification of the time series. In their paper, several experiments were done to perform classification at each time point using shorter time series. Their model was not patient-specific since the length of segments considered was very inflexible; that is, if the model is trained on time series of length 3, the prediction is always done at the third time point. Our ECM model addresses this issue directly and was, to the best of our knowledge, one of the first models that is capable of patient-specific early classification of multivariate gene expression time series.

Borgwardt et al. (2006) borrowed tools from system identification to capture the 'essence' of time series of gene expression profiles, on top of which they build an SVM kernel to discriminate between time series. Their method also takes into account the dynamic evolution over time as well as the temporal characteristics of the data. More specifically, they model the evolution of the gene expression profiles as an LTI dynamical system (Kalman filter) and estimate its model parameters. Based on those model parameters, they generate infinite sequences out of relatively short input time series. An SVM kernel out of infinite sequences is then used to classify these time series. Basic differences between our methods, besides using Kalman Filter instead of HMM, include the fact that Borgwardt et al. used transformed full time series as input into SVM, while we are using shorter segments. The other more important issue is that our algorithms are designed for early classification, while Borgwardt et al. similarly to Lin et al. model was very inflexible, i.e. for the model trained on time series of length 3, the prediction is always done at the 3rd time point.

While the task of early classification is by no means new, its application to multivariate time series is an extremely unexplored field. In fact, to the best of our knowledge, the only other research in this area was performed by Ghalwash and Obradovic (2012).

Ghalwash and Obradovic (2012), in their method MSD (Multivariate Shapelet Detection) adapt the use of time series segments called shapelets, which were originally developed for univariate time series by Xing et al. (2011, 2012), for early classification of multivariate data. Time series patterns were extracted from all dimensions of the time series that distinctly manifest the target class locally. The time series were classified by searching for the earliest closest patterns. While this shapelet-based approach was able to outperform a number of baseline models, it had several limitations: the components of the multivariate time-series shapelet needed to have the same starting positions and the MSD method could not handle missing values. Moreover, MSD was unable to address the varied response rate problem that is often encountered in clinical data.

Our ECM model uses HMMs to transform short segments of time series into their log likelihoods, which are then used as input to SVM, in order to obtain predictions optimised for early classification. In our preliminary work which was published in BIBM conference, ECM model was tested on a published dataset of response to drug therapy in Multiple Sclerosis patients, used only an average of 40% of a time series and was able to outperform some of the baseline models, which needed the full time series for classification (Ghalwash et al., 2012).

2 Model description

In this study, we propose using a hybrid model that combines a generative model (HMM) with a discriminative model (SVM) for early classification of a multivariate time series. We first give a brief introduction to HMM and SVM and then introduce our hybrid model in the following sections.

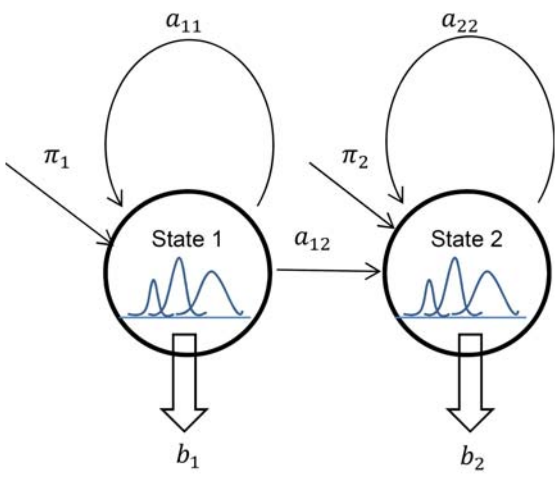
2.1 HMM

Hidden Markov Models (HMMs) are powerful statistical models for modeling time-series data that can be characterised by an underlying process generating an observable time series (Rabiner and Juang, 1986; Rabiner, 1989). The HMM, referred to as Λ, is defined as tuple $\Lambda = (S, \Pi, A, B)$ where

- S is a finite set of states.
- $\Pi = \{\pi_i\}$ are the initial state probabilities.
- $A = \{a_{ij}\}$ are the state transition probabilities, where a_{ij} is the probability of transition from state i to state j.
- $B = \{b_i\}$ is the emission probabilities, where b_i is the probability of generating a real-value at state i.

Lin et al. (2008) showed that using two states allows for efficient alignment of patients with different response rate, as the two states represent two phases of the response (phase or disease in our context). Therefore, in our context we assume that the cardinality of the set of states S is 2, see Figure 2. The emission probability is assumed to be drawn from a multivariate Gaussian distribution whose dimension equals the number of genes. In other words, $b_i \sim \mathcal{N}(\mu_i, \sigma_i)$ where μ_i and σ_i are the mean and standard deviation of the multivariate Gaussian distribution of the state i. Hence, the covariance matrix is assumed to be diagonal to avoid overfitting and to allow for fast computation of the model.

Figure 2 HMM architecture (see online version for colours)



Notes: π_i is the initial state probability. a_{ij} is the transition probability from state i to state j. b_i is the emission state probability which is drawn from a mulitvariate Gaussian distribution of the state i.

2.2 SVM

SVMs belong to the family of kernel-based techniques that have proven to be extremely effective for machine learning purposes. They are a popular tool as they have a number of useful features, such as having theoretical guarantees of performance and not being affected by the curse of dimensionality. In instances where the data is not linearly separable, SVMs can be used to transform the dataset by mapping it to a high-dimensional feature space via the kernel trick (Boser et al., 1992). After this transformation has been performed, the SVM finds the maximal margin hyper-plane, expressed as a linear combination of support vectors of the training set, which is then used to classify the data:

$$f(\mathbf{x}) = sgn(\sum_{i}^{M} y_i \alpha_i \phi(\mathbf{x_i x} + \delta)) \tag{1}$$

The set of parameters α_i is computed by solving a convex quadratic programming problem with linear constraints (Burges, 1998). Several kinds of kernel functions are often used with SVMs, including polynomial kernels, Radial Basis Function (hereinafter RBF) kernels and sigmoid kernels. δ can be obtained using the Karush-Kuhn-Tucker condition (Burges, 1998).

2.3 Hybrid model for early classification

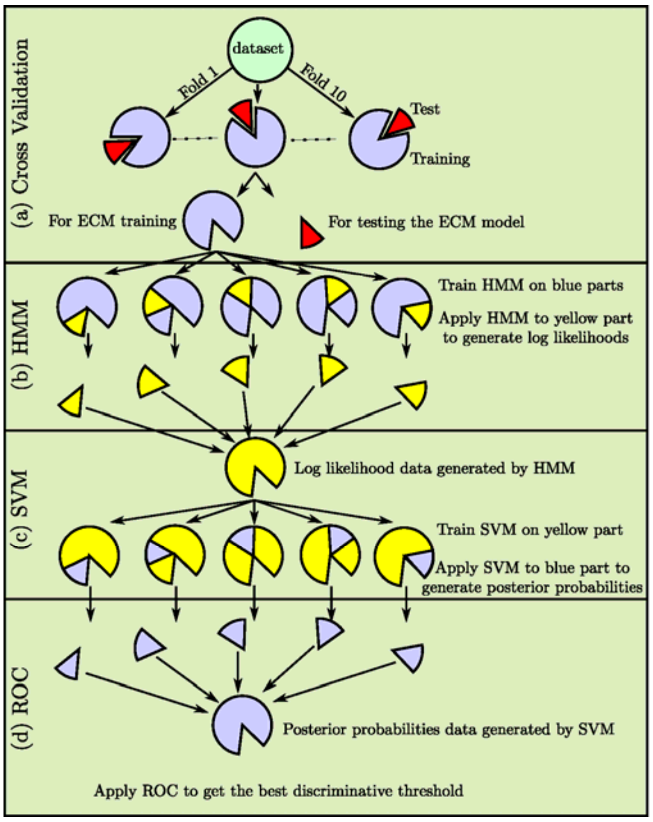
The HMM Λ is used to learn the distribution of the patterns in a time series gene expression training dataset. The SVM takes the outputs of the HMM models as input: log likelihoods of segments in the time series gene expression training dataset. The role of SVM is to determine the combination of HMM models that could be trusted when classifying the time series based on the current stream.

The reasoning behind using a hybrid model is that if a generative model encounters a short segment that is repeated often in both classes (which can be a fairly common occurrence in gene expression data), it will likely not be able to distinguish between the two classes effectively. By applying a discriminative model on top of the generative model's output, we are able to overcome this problem and improve overall performance. The details of how the hybrid model is trained and applied to the test data are provided next.

2.3.1 Data preparation

The dataset is first divided into training and test data. The test data remains untouched until the very end and all of the following procedures are applied to only the training data (see Figure 3 (a)). When we mention test time series in the Training Phase, we are referring to the portion of the training data held out for testing for a particular fold of the internal cross-validation.

Figure 3 ECM training process: (a) the dataset is divided into ten partitions for ten cross-validation process. For each fold, ECM is trained on nine partitions and then tested on one partition; (b) for each training data, the data is divided into five partitions where HMM is trained on four partitions and applied on one partition to generate log likelihood data for SVM; (c) the log likelihood data is divided (same partitions as in HMM) into five partitions where SVM is trained on four partitions and applied on one partition to generate the posterior probabilties; (d) the ROC curve approach is applied to the posterior probabilities data to identify the best discriminative margin threshold for SVM (see online version for colours)


(a) Cross Validation
dataset
Fold 1
Fold 10
Test
Training
For ECM training
For testing the ECM model
(b) HMM
Train HMM on blue parts
Apply HMM to yellow part to generate log likelihoods
(c) SVM
Log likelihood data generated by HMM
Train SVM on yellow part
Apply SVM to blue part to generate posterior probabilities
(d) ROC
Posterior probabilities data generated by SVM
Apply ROC to get the best discriminative threshold

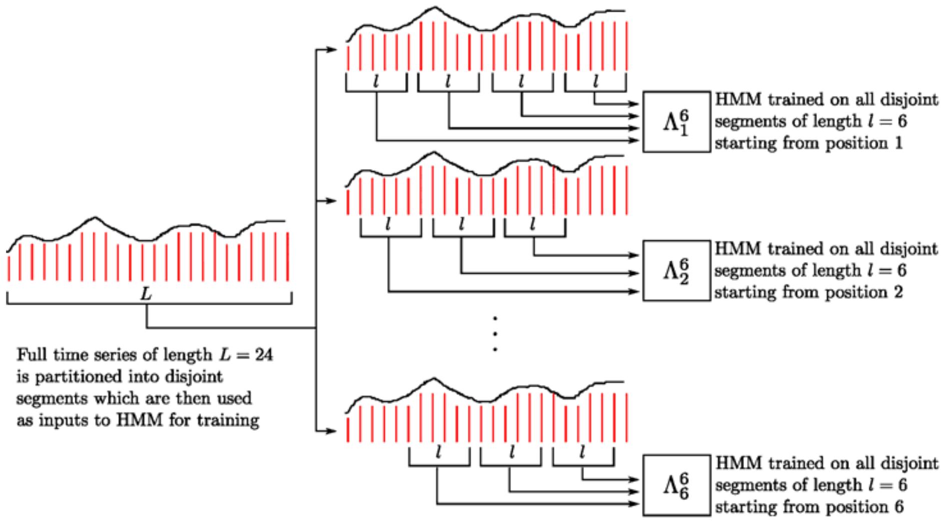
2.3.2 Training phase

In general, several HMMs are trained on short time series segments and their log likelihoods on validation data are generated as features for the SVM. We opted for using an RBF kernel SVM.

STEP1: Train HMMs of specific length

Given a time series of length L from one class, we extract all disjoint segments of length l, starting from position 0 and an HMM is trained on those segments. We refer to that HMM as Λ_1^l. We then shift the starting position by one time point, extract all disjoint segments of length l starting from position 1 and train another HMM on those segments. We call it Λ_2^l. If $l \leq L/2$, we generate l HMM models. We vary the length and the appropriate shifts in order to capture every possible pattern, as in Figure 4, since these patterns can have different lengths and start at different positions.

Figure 4 Step 1 of the algorithm is explained on the time series of length L = 24 and for HMM trained on all segments of length l = 6 (see online version for colours)



L
Full time series of length $L = 24$ is partitioned into disjoint segments which are then used as inputs to HMM for training
l
Λ_1^6
HMM trained on all disjoint segments of length $l = 6$ starting from position 1
Λ_2^6
HMM trained on all disjoint segments of length $l = 6$ starting from position 2
Λ_6^6
HMM trained on all disjoint segments of length $l = 6$ starting from position 6

Therefore, the number N_l of the HMM models, Λ_i^l $where i = \{1, \ldots, N_l\}$, trained on all segments of length l at every starting position is

$$N_l = \begin{cases} l & if l \leq L/2, \\ L - l + 1 & otherwise. \end{cases}$$

STEP2: Train HMMs for different lengths

We repeat the above procedure for different segment lengths l up to $l + k$. k is a user-defined parameter, set small enough to be able to capture all possible patterns, but $l + k$ should not exceed 50% of the time series to still be able to identify the pattern as early as possible. Hence, the maximum number of models to be generated is

$$N_l + N_{l+1} + \ldots + N_{l+k} = l + (l+1) + \ldots + (l+k) = (k+1)(k/2 + l).$$

The same technique is applied to the time series of the other class, resulting in a maximum of N models where:

$$N = 2(k+1)(k/2+l).$$

STEP3: Generate features for SVM

We apply all trained HMMs on the test data as follows: first, we take the shortest time series segment O extracted from the training data. If the length of that segment is l, we then read l time points from the test time series (O_l). Next, we ask all N HMMs to generate their log likelihoods on the test time series we are currently examining. In general, an HMM is able to generate a log likelihood on the test time series if the model is trained on segments of length shorter than or equal to the length of the test time series. If the length of the current test time series is greater than l, then the Λ_i^l model uses only the last portion l time points of the current time series, such that the length of this portion has the same length as the segments used for training the HMMs. The log likelihoods generated in this fashion, $\log P(O_l|\Lambda_j); j = \{1,\ldots,N\}$, are then considered as features for an SVM (see Figure 3 (b)).

The next part of this step is to read one more time point from the time series, so that the current test time series length is incremented by one. All models are asked to provide their log likelihoods on this time series to be another row for the SVM. This process is repeated until we have read all of the time points of the test time series. After this stage, we will have generated N_l instances with dimensionality N for the SVM, all of which have the class label of the test time series. Finally, we repeat this procedure for all test time series.

We repeat steps 1–3 five times to obtain five-cross validations. Here we reinforce the fact that the HMMs are not used for prediction of the class of the time series, but rather their likelihoods are used as data for the SVM, which is made aware of the class labels independent of the HMMs.

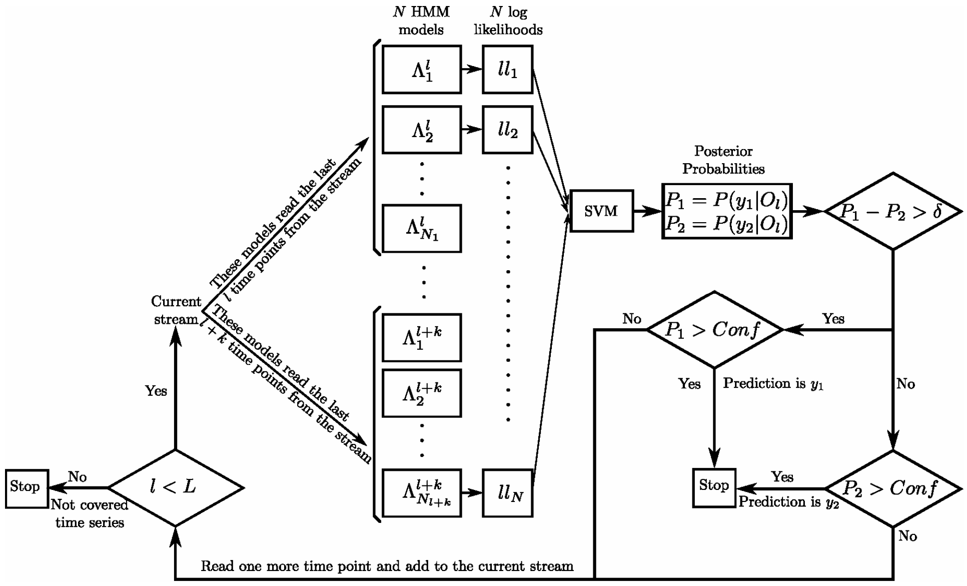
After the cross validation is done, the log likelihoods generated by HMMs are ready to be used as features for the SVM. The parameters of the SVM are optimised using cross validation on the log likelihoods generated by the HMMs. Then another five-cross validation is conducted for SVM to generate a score for each example. We use the same partitions of the five-cross validations used for the HMM training phase to avoid any bias (see Figure 3 (c)). When testing the SVM, the model computes a posterior probability estimate $P(y|O_l)$ of the class y for each example O_l. Namely, in a binary classification context, an SVM gives two probabilities: $P(y_1|O_l)$ and $P(y_2|O_l)$ (note that $P(y_2|O_l) = 1 - P(y_1|O_l)$). The scores for all examples are computed as the difference between the two probabilities $P(y_1|O_l) - P(y_2|O_l)$. Using the scores of all examples and their labels, we identify (using a standard ROC curve approach) the best threshold δ that maximises the balanced accuracy (average between sensitivity and specificity) (see Figure 3 (d)). Finally, both HMM and SVM are trained on training data which although being the same is essentially different – HMMs have time series as input, while SVM has log likelihoods, the output of HMMs, as input.

2.3.3 Testing phase

We note here that the test data allocated in the very beginning of the procedure is never used for training either the SVM nor HMM models (see Figure 3 (a)).

The sketch for the test phase is illustrated in Figure 5. For a given time series with an unknown label, we read l time points (*the current stream*) from the test time series. We then ask all N HMM models to provide log likelihoods. We use their outputs as inputs to the SVM, as described above. The score of the current time series is then computed using the probabilities generated by the SVM as $P(y_1|O_l)-P(y_2|O_l)$. If the score is greater than the threshold δ, then the current time series is most likely from class y_1, otherwise, it is from class y_2. We then ask if the probability estimate for the selected class is high enough to be confident about the prediction. A user-defined parameter *Conf* is used to measure this level of confidence. If the probability estimate is higher than the confidence level *Conf*, we stop at the current time point and predict the time series. Otherwise, we read one more time point from the current time series and repeat the procedure. If we reach the end of the time series and are not able to classify the time series, we mark that test time series as 'not covered time series'.

Figure 5 ECM Test phase



Notes: Each HMM model Λ_i^l reads the last l time points from the current stream and generates log likelihood ll. The log likelihoods are then passed to SVM to generate the poseterior probabilities of the time series being generated from each class. The margin between the two probabilities is compared to the threshold δ to determine the class membership. If the probability of the predicted class is greater than the confidence level *Conf* then the process is stopped. Otherwise, we read one more time point and then the process is repeated until the prediction is obtained or we reach the end of the time series.

2.3.4 Whole process

The above training and testing phases are repeated 10 times using 10 different partitions (10 cross validations). The average accuracy, as well as the coverage rate (how many time series out of the test time series data were classified) and the fraction of the time series used for classification, are reported.

3 Experiments

3.1 Experimental setup

All of the procedures have been implemented in C++. We used libSVM, which provides probability estimates (Chang and Lin, 2011). The code is publically available at www.dabi.temple.edu/~mohamed/ECM. All experiments are conducted on a PC Intel Core i7 2.8 GHz with 8GB RAM.

3.2 Dataset description and results

Time series gene expression experiments have been used in a variety of biomedical applications (Tchagang et al., 2009). These experiments consist of building expression profiles, which are essentially functions that model the changes in the expression levels of various genes. Such changes are collected by performing multiple microarray experiments over a period of time (Spellman et al., 1998). Each experiment usually measures the expression level of multiple genes at a given time point. In this context, the gene expression time series classification problem is the process of determining the class of a previously unseen time series, based on the expression levels recorded for the time series in the training dataset. These classes can be defined in a variety of ways to help answer different kinds of questions. For example, as noted by Borgwardt et al. (2006), '*Will patient X respond well to a certain therapy or drug treatment? Is patient X going to develop sepsis in the next few hours? Is patient X recovering from a disease?*'

A clinical dataset, which we will refer to as MS70, was generated to study the changes in cellular functions in multiple sclerosis patients in response to drug therapy with IFNβ (Baranzini et al., 2005). The dataset contains time series gene expression values for 52 patients. The patients were classified as good responders (33 patients) or bad responders (19 patients) to the drug. Blood samples were taken every three months in the first year and every six months in the second year. Some patients miss certain measurements, especially at the seventh time point. Thus, the gene expression values were measured an average of five–seven times for each subject. In order to adhere to the limitations of clinical settings (in which only a small, pre-specified number of genes is provided) and to be able to effectively compare ECM results with those attained in other studies, several datasets comprised of a fairly small number of genes were generated.

The identification of triplets of genes for a Bayes classifier of time series gene expression data of multiple sclerosis patients' response to a drug was performed by Baranzini et al. (2005). Previous research identified 12 genes in terms of triplets. Hence, we generated four datasets: Baranzini3A and Baranzini3B, which consist of one triplet of the best two triplets of genes; Baranzini6, which has the top two triplets; and Baranzini12, which has all 12 genes identified by all triplets. A discriminative HMM has been developed and applied to the MS70 dataset to reveal the genes that are associated with the good or bad responders to the therapy (Lin et al., 2008). A total of nine genes were found that are associated with the therapy, seven of which are identified using the last time stamp. Hence, we constructed two datasets, called Lin9 and Lin7, consisting of nine and seven genes, respectively. The entire MS70 dataset, containing all 70 genes, was also used for our experiments. A mixture of HMMs has been developed to identify the genes that are associated with patient response to the treatment (Costa et al., 2009). A total of 17 relevant genes were found, so we constructed a dataset called Costa17 that is comprised of the 17 genes. Table 1 contains the list of the genes used in our experiments for the drug response datasets.

Table 1 The list of the genes used in our experiments for the drug response datasets

Dataset	*Genes*
Baranzini3A	Caspase 2, Caspase 10, FLIP
Baranzini3B	Caspase 2, Caspase 3 , IRF4
Baranzini6	Caspase 7, Caspase 10, IRF2, IRF4, IRF6, IL-4Ra
Baranzini12	Caspase 2, Caspase 3, Caspase 7, Caspase 10 Flip, IRF2, IRF4, IRF6, IL-4Ra, IL12Rb1, STAT4, MAP3K1
Lin9	Caspase 2, Caspase 3, Caspase 10, IL-4Ra IL12Rb2, MAP3K1, IRF8, Jak2, RAIDD
Lin7	Caspase 2, Caspase 3, Caspase 10 IL-4Ra, MAP3K1, Jak2, RAIDD
Costa17	Caspase 2, Caspase 3, Caspase 10, Caspase 5 MAP3K1, STAT4, IRF2, IRF4, IRF5, IRF8, BAX, Tyk2 IL-4Ra, IL-2Rg, IFN-gRb, IFNaR2, Jak2

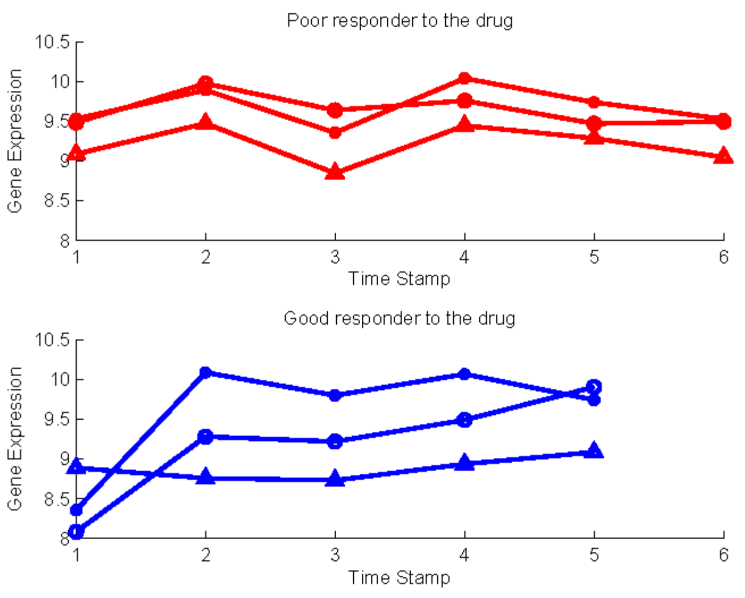
In order to have a possibility of effectively comparing our method with other methods proposed in literature, we first created 8 datasets that posses the same characteristics as those presented by Baranzini et al. (2005), Lin et al. (2008) and Costa et al. (2009). Three of these datasets were used to determine the best confidence value to be used by our model, since it was the user-defined parameter in our method. In this set of experiments, we trained our model with a distinct value for the confidence parameter (0.4, 0.5 and 0.6) on Baranzini6, Baranzini12 and Lin9, respectively. Note that although the possible values of the confidence parameter are in the [0, 1] range, we noticed that accuracy drops for values of confidence parameter smaller than 0.4 and coverage suddenly drops when we choose values greater than 0.6. Both trends could be easily explained: if we allow decisions when we are not confident, our model will decide early, but the accuracy will not be sufficient. On the other hand, if our requests for confidence are too high, we have generally higher relative accuracy scores, at the expense of coverage and earliness and we might end up without decisions until the last time stamp. Therefore, we omitted values below 0.4 and above 0.6, for the set of experiments dealing with gene expression classification. Finally, we opted to use a value of 0.5 for the confidence parameter, as this value seemed to offer the best middle-ground of the various tradeoffs evident in Table 2.

Table 2 Classification accuracy, coverage and earliness in prediction at different levels of confidence (*Conf*) on various datasets

		Baranzini6	*Baranzini12*	*Lin9*
	Conf = 0.4	62	85	87
Relative accuracy	*Conf* = 0.5	66	85	85
	Conf = 0.6	71	83	86
	Conf = 0.4	96	100	100
Coverage	*Conf* = 0.5	92	100	100
	Conf = 0.6	86	98	94
	Conf = 0.4	44	43	43
Earliness	*Conf* = 0.5	49	43	44
	Conf = 0.6	51	46	50

To further explain the effectiveness of our approach we show a real case from the Baranzini3A dataset. Figure 6 illustrates a 3-dimensional gene expression time series (genes Caspase 2, Caspase 10 and FLIP) observed at six time steps. In the top panel, a patient who is a poor responder to the drug is correctly classified by our hybrid approach at the third time point. In the bottom panel, a patient who is good responder to the drug is correctly classified at the fourth time point. Although the patterns of both patients look similar to each other such that it is hard to distinguish between them by eye, our hybrid model was able to classify them correctly and early.

Figure 6 Time series for three gene expression for a patient who is a poor responder (top panel) and good responder (bottom panel) to the drug are represented (see online version for colours)



Notes: The subjects have been correctly classified at the third and fourth time point, respectively.

Figure 7 compares different values for the user parameter confidence, which measures the confidence level the user expects from the model. Since the relative accuracy is barely affected by the value of confidence (in the specified 0.4–0.6 range) over the eight datasets, it is clear that the model is insentitive to the parameter and the results are fairly robust. The same comparison has been conducted to see the effect of different values of the confidence parameter on the earliness measure, as shown in Figure 8.

Figure 7 Measuring the sensitivity of the user parameter confidence on the relative accuracy for different datasets (see online version for colours)

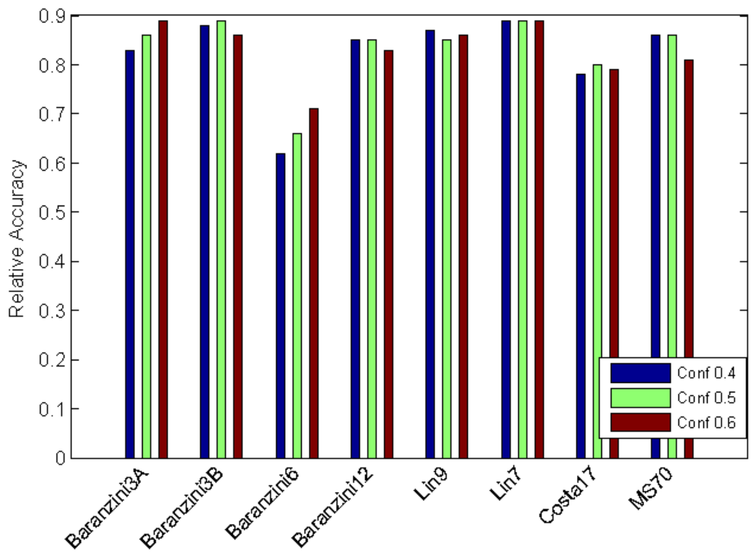
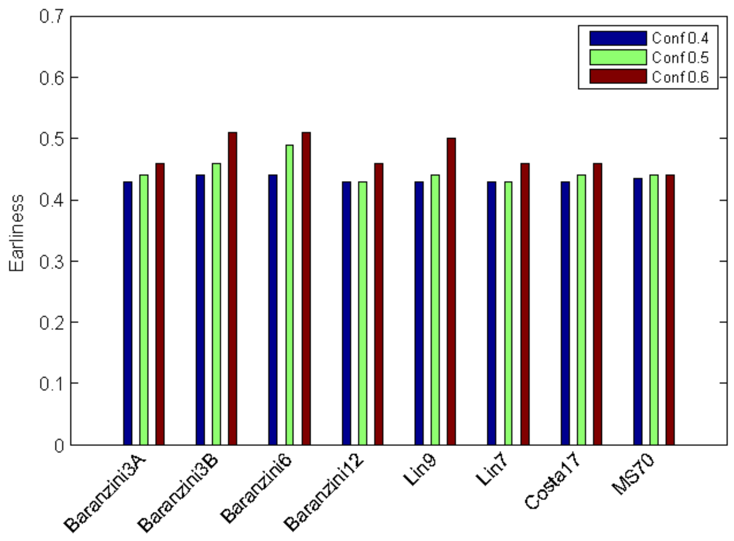


Figure 8 Measuring the sensitivity of the user parameter confidence on the earliness for different datasets (see online version for colours)



The outcome of applying our model to the remaining 5 datasets (out of the 8 that we initially created) is shown in Table 3. The relative accuracy of our model on each of the datasets is reported, along with the performance of the best models presented in literature when applied to the same datasets. Note that the accuracy of the 'best' models is based on using the entire time series, while the accuracy of our model is based on using, on average, only 3/7 of the testing time series and obtaining patient-specific decisions. This important fact emphasises the effectiveness of our method: even though the best models on average outperform our model in terms of accuracy, we are able to provide competitive accuracy scores while using a significantly smaller portion of the full time series. In two of the datasets we were actually able to outperform the best models on accuracy, in spite of utilising less than a full time-series when testing, reinforcing our model's efficiency. The performance of MSD on all of these datasets is also reported, although it performed rather poorly across the board and could not even handle the MS70 dataset. This was most likely due to the high amount of missing values in these datasets, since MSD applies cut and glue strategy in the case of missing values.

Table 3 Classification accuracy and earliness in prediction (in parentheses) for MSD, the best model in literature and our approach on various datasets

Dataset	*MSD*	*Best*	*ECM*	*Better?*
Baranzini3A	72.7 (~3/7)	87.8 (7/7)	86.2 (~3/7)	×
Baranzini3B	65.2 (~3/7)	87.5 (7/7)	88.5 (~3/7)	√
Lin7	70.8 (~4/7)	85.0 (7/7)	88.7(~3/7)	√
Costa17	66.7(~3/7)	92.7 (7/7)	79.7 (~3/7)	×
MS70		81.4 (7/7)	72.0 (~3/7)	×

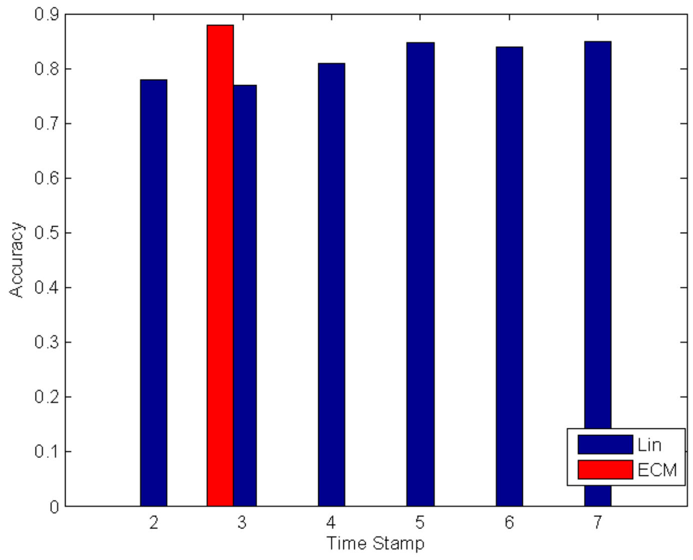
Notes: ~ means on average. Wherever our approach achieved higher accuracy column Better? has a √ mark, otherwise there is a × mark.

Figure 9, which provides the accuracy of the model developed by Lin et al. (2008) when using a shorter time series of fixed length in training and testing, further displays the strength of our model, considering that the only other method capable of early classification in a multivariate context was noticeably less accurate.

3.3 *Sepsis therapy dataset results*

To test the performance of ECM on a sepsis-related dataset we opted for a sepsis model of the acute inflammatory response dataset, which consists of four variables varying over time (Day et al., 2010). The authors adopted a subsystem approach to ensure that the interactions of the model variables are consistent with biological observations. Therefore, the model is able to capture a variety of clinically relevant scenarios associated with the inflammatory response to infection. The model displays three physiologically relevant equilibrium points, which correspond to biological states of health, aseptic death and septic death. We chose to use this model since, to the best of our knowledge, there is currently no publically available clinical dataset possessing the characteristics necessary for our experiments and we used this model to generate a small number (50) of patients to further adhere to the restrictions of a real-world scenario.

Figure 9 Comparison of the accuracy when using a shorter time series of fixed length in training and testing (Lin's model) and ECM (different segment lengths and different shifts) on Lin7 dataset (see online version for colours)



Notes: ECM accuracy at the average earliness, consisting of three time points, is shown. Lin's results on all time stamps were worse than ECM results.

We first obtained the relevant statistics of the threshold-based approach that is used for examining patients in hospitals. This approach relies on monitoring the value of one characteristic variable and labelling the patient as non-healthy as soon as it passes the threshold of 0.05. Although this is a fairly simple approach, it is currently considered to be the state-of-the-art and is used in hospitals. However, the approach of this model is too aggressive and even patients who do not need therapy are treated. To test this model in comparison to ECM, a population of 50 virtual patients is generated by the mathematical model (Day et al., 2010) on the simulation time of 24 hours (one day) with hourly observations of all 4 given variables. Virtual patient state was determined from the values of the variables at the end of the total simulated time spent in the hospital. The population consisted of 29 virtual patients classified as healthy and 21 classified as septic or aseptic (needed treatment). We also use this generated data to test the performance of MSD, which was able to perform much better on this dataset than on the drug response datasets since there were no missing values. The results are shown in Table 4 and it is evident that:

- The threshold and MSD methods appropriately treat unhealthy patients in the second or third hour.
- Our approach appropriately treats the unhealthy patients in exactly the second hour.
- At the same time, healthy patients are not treated at all until the model confirms that they do not need treatment (which occurs by the fourth hour at the latest), at which point the remainder of the time series no longer needs to be considered.

- Both the threshold and our method do not generate any false negatives, which is a vital statistic in the medical field (every false negative is essentially a patient that ends up dying due to misclassification).
- Although MSD offers very similar results to our approach, it did result in one false negative, which is a major limitation in clinical applications.

Table 4 Classification results of threshold-based method, ECMTS and ECM on virtual patient data

Model	*FN*	*FP*	*TP*	*TN*	*Earliness*
Threshold	0	7	21	22	2–3
MSD	1	3	20	26	2.12
ECM	0	3	21	26	2

The effectiveness of our method becomes clear when examining the false positive and earliness scores: using our method, four virtual patients were spared an incorrect diagnosis of sepsis. In reality that would effectively save real patients money on treatment and preserve their health (since unnecessary treatment can cause damage and eventually death). Furthermore, both the threshold-based method and MSD predicted all of the unhealthy patients at either the second or the third time step, whereas our method was able to do so without needing the third time step. Finally, while the performance of MSD was very similar to our approach, the false negative it produced suggests that our method is better.

4 Conclusion

In this paper we proposed a novel application of a hybrid HMM and SVM model in the realm of biomedical data. Our ECM is able to handle multivariate gene expression time series data and is capable of performing patient-specific early classification on this kind of data with unprecedented results. On datasets gathered from real-world scenarios, our method achieved relative accuracy scores that were on average slightly lower than the best results presented in literature when models were tested on the same datasets (although on two of these datasets, our model had a higher accuracy score). However, while the models from literature attained their best results after observing the full time series, our model was able to keep up by utilising only 40% of the whole time series, thus displaying its early classification potential. Our approach was also able to significantly outperform the only other method designed specifically for early classification of multivariate time series. The performance of our model on a set of virtual patients was even more impressive: when compared to results obtained by applying the decision criteria used in the vast majority of hospitals, our model was able to correctly classify a larger proportion of patients, without misclassifying (and thus killing) any patients. Furthermore, it was able to do so as early or even earlier than both the standard threshold-based approach and MSD, without generating any false negatives, suggesting that there are no downsides in using our method over the threshold-based approach or MSD for this purpose.

In closing, one of the assumptions of using HMMs is that there are enough examples available for the model to accomplish a sufficient amount of learning. Because medical data is often noisy and expensive to obtain, this assumption cannot always be met when real data is involved (in some real-life scenarios, the data contains only 10 or 15 examples), rendering our ECM approach useless. We intend to address this issue in the future, while still allowing for multivariate time series to be classified early.

Acknowledgements

Thanks to Alexey Uversky for proof reading. This work was funded, in part, by DARPA grant [DARPA-N66001-11-1-4183] negotiated by SSC Pacific grant; and the Egyptian Ministry of Higher Education.

References

Baranzini, S.E., Mousavi, P., Rio, J., Caillier, S., Stillman, A., Villoslada, P., Wyatt, M.M., Comabella, M., Greller, L., Somogyi, R., Montalban, X. and Oksenberg, J. (2005) 'Transcription-based prediction of response to IFNβ using supervised computational methods', *PLoS Biology*, Vol. 3, No. 1, pp.166–176.

Borgwardt, K.M., Vishwanathan, S. and Kriegel, H-P. (2006) 'Class prediction from time series gene expression profiles using dynamical systems kernels', *Pacific Symposium on Biocomputing*, Vol. 11, pp.547–558.

Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992) 'A training algorithm for optimal margin classifiers', *Proceedings of the 5th Annual Workshop on Computational Learning Theory, COLT'92*, 27–29 July, Pittsburgh, PA, USA, pp.144–152.

Burges, C.J.C. (1998) 'A tutorial on support vector machines for pattern recognition', *Data Mining Knowledge Discovery*, Vol. 2, No. 2, pp.121–167.

Chang, C. and Lin, C. (2011) 'LIBSVM: a library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp.27:1–27:27.

Costa, I., Schnhuth, A., Hafemeister, C. and Schliep, A. (2009) 'Constrained mixture estimation for analysis and robust classification of clinical time series', *Bioinformatics*, Vol. 25, No. 12, pp.i6–i14.

Day, J., Rubin, J. and Clermont, G. (2010) 'Using nonlinear model predictive control to find optimal therapeutic strategies to modulate inflammation', *Mathematical Biosciences and Engineering*, Vol. 7, No. 4, pp.739–763.

Dellinger, R., Levy, M., Carlet, J., Bion, J., Parker, M., Jaeschke, R., Reinhart, K., Angus, D., Brun-Buisson, C., Beale, R., Calandra, T., Dhainaut, J-F., Gerlach, H., Harvey, M., Marini, J., Marshall, J., Ranieri, M., Ramsay, G., Sevransky, J., Thompson, B., Townsend, S., Vender, J., Zimmerman, J. and Vincent, J-L. (2008) 'Surviving sepsis campaign: international guidelines for management of severe sepsis and septic shock', *Critical Care Medicine*, Vol. 36, No. 1, pp.296–327.

Ghalwash, M. and Obradovic, Z. (2012) 'Early classification of multivariate temporal observations by extraction of interpretable shapelets', *BMC Bioinformatics*, Vol. 13, p.195.

Ghalwash, M.F., Ramljak, D. and Obradovic, Z. (2012) 'Early classification of multivariate time series using a hybrid HMM/SVM model', *IEEE International Conference on Bioinformatics and Biomedicine*, 4–7 October, Philadelphia, PA, USA, pp.1–6.

Huang, B.Q., Du, C.J., Zhang, Y.B. and Kechadi, M-T. (2006) 'A hybrid HMM-SVM method for online handwriting symbol recognition', *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications, ISDA'06*, 16–18 October, Jinan, China, pp.887–891.

Jiang, H. (2010) 'Discriminative training of HMMs for automatic speech recognition: a survey', *Computer Speech and Language*, Vol. 24, No. 4, pp.589–608.

Lin, T., Kaminski, N. and Bar-Joseph, Z. (2008) 'Alignment and classification of time series gene expression in clinical studies', *Bioinformatics*, Vol. 24, No. 13, pp.i147–i155.

Rabiner, L. (1989) 'A tutorial on HMM and selected applications in speech recognition', *Proceedings of the IEEE*, Vol. 77, No. 2, pp.257–286.

Rabiner, L.R. and Juang, B.H. (1986) 'An introduction to hidden Markov models', *IEEE ASSP Magazine*, Vol. 3, No. 1, pp.4–16.

Schliep, A., Schönhuth, A. and Steinhoff, C. (2003) 'Using hidden Markov models to analyze gene expression time course data', *Bioinformatics*, Vol. 19, No. 1, pp.i255–i263.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) 'Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization', *Molecular Biology of the Cell*, Vol. 9, pp.3273–3297.

Tchagang, A.B., Bui, K.V., McGinnis, T. and Benos, P.V. (2009) 'Extracting biologically significant patterns from short time series gene expression data', *BMC Bioinformatics*, Vol. 10, p.255.

Xing, Z., Pei, J. and Yu, P.S. (2012) 'Early classification on time series', *Knowledge and Information Systems*, Vol. 31, No. 1, pp.105–127.

Xing, Z., Pei, J., Yu, P.S. and Wang, K. (2011) 'Extracting interpretable features for early classification on time series', *Proceedings of 11th SIAM International Conference on Data Mining*, 28–30 April, Mesa, AZ, USA, pp.439–451.