Piecewise Aggregation for HMM fitting. A pre-fitting model for seamless integration with time series data.

Joaquim Assunção [†], Jean-Marc Vincent [*], Paulo Fernandes [‡]

[†] UFSM - Department of Applied Computing - Santa Maria, Brazil
[*] Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
[‡] Roberts Wesleyan College – Rochester, NY – USA
[†] joaquim@inf.ufsm.br, [*] jean-marc.Vincent@imag.fr, [‡] fernandes_paulo@roberts.edu

Abstract

Broadly used and applied in many domains, Hidden Markov Models are a well established formalism, both in computer science and statistics. Among other reasons, they owe their popularity to a fast fitting method, *i.e.*, the Baum-Welch algorithm, allowing to adjust models to a variety of input data. Using expectation and maximization phases, BW assures an increase to the model likelihood at every iteration. Yet, to initialize the sequence of expectation-maximization (EM) steps, it is a standard procedure to start the BW algorithm from randomly generated values. We propose a, simple and fast, deterministic pre-fitting approach which derives the BW's initial values directly from the input data.

1 Introduction

Due to their relative simplicity and power to represent complex systems, Hidden Markov Models (HMMs) are one of the most widely used stochastic formalism for time series [31]. HMMs owe their flexibility and their ease of application to a well-developed methodological framework, including a model fitting algorithm. The so-called Baum-Welch algorithm (BW) can be considered a special case of the Expectation-Maximization (EM) algorithm (Section 2). In essence, it derives the maximum likelihood estimates for the parameters of a given model, *i.e.*, the best fit for the input data in the sense that the probability to observe the data given the model parameters is maximal. This is achieved by an iterative procedure guaranteed to increase the value of the likelihood at each iteration and such procedure finds a local maximum; therefore, offering a model solution even when the likelihood is untraceable or too costly to maximize directly. However, as well as EM algorithms in general, BW tends to be sensitive to its input parameters [6][9].

Several extensions have been suggested to overcome the flaws of EM [9][18][13][29]. They are based on combinations of algorithms and techniques such as classification, randomization or more complex stochastic additions. Others, focused on improving the BW, have different approaches performing changes within the algorithm, which can also deal with possible convergence problems [4][26][16] [20].

Several extensions have been suggested to overcome the flaws of EM [9][18][13][29]. They are based on combinations of algorithms and techniques such as classification, randomization or more complex stochastic additions. Others, focused on improving the BW, have different approaches performing changes within the algorithm, which can also deal with possible convergence problems [4][26][16] [20].

Our solution focuses on the initialization only. It keeps the BW's structure and adds a pre-fitting deterministic step, which by avoiding bad initial parameters tends to obtain higher likelihood values in the first iteration, thus reducing the number of iterations needed to find a local maximum, which leads to a fast model fitting (Section 3). Although the difference is minimal, the possibility of deriving the initial values directly from the input data can be interesting for applying these models into different scenarios.

Since an HMM is usually used having serial data as its input [31], the idea is to use a deterministic discretization technique for time series prior to the actual model fitting. From this approximate description of the original observations, we then derive initial parameters which are fed into BW. These parameters are, in general, reasonably close to a local maximum. Thus, the combination of the parameter's selection step, in combination with the BW algorithm, is more likely to reduce the number of iterations to maximize the parameters, therefore leading to a local maximum likelihood faster than the most traditional approach, which is using random numbers to initialize the parameters.

Our approach is based on a Piecewise Aggregate Approximation (PAA) technique, which is used in the algorithm (Figure 1, Piecewise Expectation). The use of PAA with EM, Piecewise Aggregation EM (PAEM), has advantages regarding fitting speed and practicality due to its simpler set of parameters.

[*] DOI reference number: 10.18293/SEKE2019-185

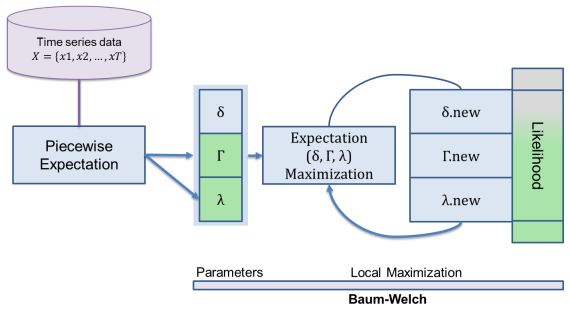

Figure 1: PAEM, schematic representation

Finally, we performed experiments and measures to compare the traditional use of BW against PAEM (Section 5). The main contribution of our approach is to reduce the time needed to fit HMM models. However, this is performed with a pre-fitting without modifying the original algorithm, which is a different approach than the others (Section 2.1). This pre-fitting can also be seamlessly used on time series data, reducing the modeling time.

2 Baum-Welch

The Baum-Welch algorithm (BW) is a version of EM algorithm for HMMs [5]. The goal is to estimate the parameters of an HMM given an input data [31], in the statistics literature, commonly described as observations. This goal implies that unlike a manual fitting approach, the initial distribution δ, the transition probability matrix Γ and the emission probabilities λ, are not estimated by the modeler observations but, automatically, by the model itself.

The BW algorithm works iteratively by successive maximizing local approximations of the likelihood function. It is guaranteed to maximize the likelihood at each iteration. EM alternates between two steps. The E-step computes the conditional expectation of the hidden states given the observations and the model current parameters, Γ, δ and λ. These computations are based on the complete data log-likelihood, which is basically the natural logarithm of the likelihood function to avoid the underflow problem [14]. In the M-step, the expectations are maximized according to its parameters.

2.1 Variants and derivations

There are several variants and extension for the standard EM algorithm, also described as Generic EM (GEM). This section concisely shows an extensive literature review for the EM variants and derivations. We briefly classify the GEM-based algorithm, in order to establish their relations and differences compared to our approach.

We can globally classify these variants into deterministic and stochastic versions. Among the deterministic versions, Classification EM (CEM) [9], Accelerated EM (AEM) [18], Aitken's accleration (AA)[13], [25], Expectation Conditional Maximization (ECM)[29], ECM Either (ECME) [23], Space-Alternating Generalized EM (SAGE) [15], Parameter-Expanded EM (PX-EM) [24].

The stochastic versions include Stochastic EM (SEM) [8], Stochastic Approximation type EM (SAEM) [8], Data Augmentation algorithm (DA) [27] and Monte Carlo EM (MCEM) [30]. Although many of them are focused on Gaussian mixture models, all these variants have slightly different approaches to solving slightly different problems. A common problem is the EM step sensitiveness to the initial parameters [6]. Bad initial parameters will lead to more EM steps (iterations), which are necessary to find the local maximum likelihood.

All these GEM-based algorithms have in common the use of an iterative MLE or Recursive MLE (RMLE). However, not all fitting algorithms are based on the MLE. Some are based on Minimum Model Divergence (MMD) and Minimum Prediction Error (MPE), which can be extended to Recursive Prediction Error (RPE) as a general recursive stochastic algorithm [3]. MMD, in few words, can be described as a combination of MLE and the minimization of parameter's divergence using entropy measures. Also, Minimum Prediction Error (MPE) which consists of measuring an HMM error output prediction and provide an updated estimation for the HMM parameters [11]. Among the algorithms, using MPE we can emphasize Collings *et al.*, [11] and LeGland and Mevel [21]. Using MMD we can emphasize Garg and Warmuth [17].

Despite the uses of MMD and MPE, we focus on the classic MLE, which is commonly used for HMMs. So far, works based on MLE are the following: [4][26][16][20][7]. However, our approach does not intend to create an entirely new algorithm nor improve it within itself. Instead, we perform a pre-fitting to avoid the BW sensitiveness, an approach used by some EM variants. Also, we do not intend to create an optimal algorithm, but a better version of the traditional BW, which aims to be a practical option that does not suffer from the same flaws of a GEM, which as well as BW, is strongly dependent on its initial parameters [9]. Therefore, the convergence time is directly dependent on how good are these initial conditions.

3 PAEM

The efficiency of fitting HMMs can be improved by combining the EM algorithm with data mining techniques such as classification methods or the K-means algorithm [9][31]. These techniques are used to choose the initial parameters intelligently, thus reducing the impact of the EM/BW sensitiveness to them.

As showed in the Section 2.1, there are many algorithms which have been derived from the generic EM. However, they are based on different techniques and adapted for different situations. Here, we adapt and combine a Piecewise

Aggregation technique for time series to represent the original observations and perform a pre-fitting for the BW algorithm, calling this extension the Piecewise Aggregation EM (PAEM). PAEM profits from the simplicity of the PAA method and the dynamism of a SAX [22] inspired method, which can be applied for different kinds of distributions concerning time series. These characteristics allow us to derive meaningful initial parameters by a fast approximation of the data, avoiding failing on dimensionality problems, such as a fail to converge, which can be given by higher dimensions; or imprecise representations, which can be lead by a strong dimensionality reduction.

PAEM's initial approximation enhances the initial parameters Γ and λ, making them close to the global maximum, which leads to a faster fitting compared to the traditional random initialization. This is due to the need of a single initialization to maximize the parameters and to a first better fitting, which reduces the sensitiveness effect and tend do avoid EM iterations. Fig. 1 illustrates the general idea: a pre-fitting in a phase called piecewise expectation prior to the traditional BW. Two of three parameters are previously updated, Γ and λ, which together with the steady state, stored in δ, tends to lead to a first better likelihood. The piecewise expectation cost is $\vartheta(T)$, which is lower than forward-backward procedures $\vartheta(N^2T)$, briefly described in the previous section. Therefore, once a pre-fitting saves one iteration, the final computational cost should be lower.

3.1 Piecewise Expectation Piecewise Aggregate Approximation (PAA) is a technique to reduce data dimensionality through discretization. It has been widely applied in the context of time series analysis. Despite being simple and intuitive, PAA has been shown to be as powerful as more sophisticated dimensionality reduction techniques such as Discrete Fourier Transform [1], Discrete Wavelet Transform [10], Singular Value Decomposition [19].

To perform dimensionality reduction, PAA creates a discrete version of the original TS in w blocks. These blocks are usually a division of the length of the TS. In our case, the faster mapping characteristic is especially attractive. Since we intend to reduce the total time necessary to fit a model, more robust approaches might be too costly for a pre-fitting procedure.

Given a time series S with length n, PAA(S) is defined as a sequence PAA(S)= $\{\mu(B_1,\mu(B_2),...,\mu(B_w))\}$, where μ is the mean, w is the maximum number of blocks and B_i is a block in the index i, being ($1 \leq i \leq w$). The mean of a block is given by the Equation 3.1. If the division n/w results in a float number, the result is truncated and another block is made of the remaining part of the series.

$$\mu(B_i) = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} S_j \tag{3.1}$$

Despite its simplicity, PAA is enough to start an appropriated representation of a given series. We use it as a mapping to set up variables and then calculate Γ through MLE. Other PAA advantage is that it has only one parameter and due to it being mainly used to reduce a time series dimensionality, it is not critical for our problem. Therefore, given a set of observations $X = (x_1, x_2, ..., x_T)$, PAEM only needs one parameter, the model number of states n. It starts by getting a sequence of symbolic values ϕ, with range n, that better describes X. The total number of Blocks is equal to $w = T/m$. Thus, we identify the approximation sequence with $\phi = (\mu(B_1), \mu(B_2), ..., \mu(B_w))$.

The number of states n defines the division of the values $\mu(B_i)$. This generates the λ values of the HMM. Thus, vector ϕ has a sequence of w elements composed by n symbols. For instance, in $\phi = \{1,2,3,1,2,3,2,2\}$, $w = 8$ and $n = 3$. Now, we can extend the whole approximation process to a set of equations.

Vector ϕ is directly used to get the probabilities and set the transition probability matrix, Γ, which together with λ are the two necessary parameters for an HMM model since δ can be initiated *null* and then filled with the steady state. Due to this solution be directly related with PAA, we call this part of the algorithm, "Piecewise Expectation".

To generate Γ, we use the elements of $\phi^{(t)}$, $1 \leq t \leq w$ and $1 \leq i, j \leq n$. The non-normalized matrix Γ is filled by the cumulative sum of the probability to find an element $\phi_j^{(t)}$ just after an element $\phi_i^{(t)}$.

$$\Gamma_{(i,j)} = \sum_{t=2}^{T} P(\phi_j^{(t)} | \phi_i^{(t-1)}) \tag{3.2}$$

Different than Γ, the elements of λ are directly extracted from the piecewise approximation procedure. The generated values are actually close to the BW's ones. This small distance between the pre-fitted and the pos-fitted value is constantly observed in our experiments.

4 Experiments and Results

Our experiments aim to measure and compare how fast the local maximum likelihood is achieved through BW and PAEM. In other words, to prove the efficiency of our algorithm in finding the best model fit. To do so, we used randomly generated and randomly chosen time series from Data Market [12]. We separated our tests in three steps. First, to compare the generated maximum likelihood from different initialization of BW against the PAEM approach. Second, to detect the number of necessary executions to achieve the global maximum likelihood; consequently, how long it takes to achieve the global maximum likelihood. Third, a direct measurement of the user for PAEM vs BW.

Our tests followed the hypothesis that a human operator begins with no knowledge about the dataset. In other words,

we consider no previous data mining or machine learning techniques have been performed. For the last two sets of test (Section 4.2 and 4.3), the BW initialization followed the standard strategy [9], it was performed through random normalized numbers to all parameters. We used 12 different time series for models ranging from 2 to 4 states. Regarding these 36 tests, for each BW execution, 50 different seeds were used. To avoid outliers, the best and the worst 5 were taken out. From these 40, we used the best, the worst and the average measurements to compare against PAEM.

4.1 Likelihood Prior to the user time and iteration tests, we compared the fitness of BW and PAEM with only one initialization. Since the Expectation-Maximization part is the same, if the BW parameters are not equiprobable they should reach the same likelihood. Otherwise, if BW is inferior, it means that not all randomized parameters are good as an input. If PAEM is inferior, it means that the pre-fitting fails. Table 1 shows this experiment with BW and PAEM, where $*$ means that we used an equiprobable λ to initialize the BW. In fact, if the BW's parameters values are not equiprobable, it tends to converge to a maximum likelihood. The problem usually happens when an equiprobable λ or Γ is given as an input, which is trivial to avoid.

Table 1: Experimental model measurements using random numbers as parameter initialization.

Model	# states	mLLk	AIC	BIC
BW (Equip. λ)	2	1268.92	2547.84	2560.86
BW	2	635.32	1280.65	1293.67
PAEM	2	635.32	1280.65	1293.67
BW (Equip. λ)	3	1268.92	2559.84	2588.50
BW	3	510.22	1042.44	1071.09
PAEM	3	510.22	1042.44	1071.09
BW (Equip. λ)	4	1268.92	2575.84	2625.34
BW	4	471.82	981.65	1031.15
PAEM	4	471.82	981.65	1031.15

Although an equiprobable λ suggests a bad fitting, this is not true for all scenarios. Despite a tiny improvement, in some cases, an equiprobable λ retrieved a better likelihood. For the other datasets, a similar phenomenon occurred in some models with more than 3 states. This suggests that a simple condition to avoid an equiprobable parameter may not be a good solution.

Considering one decimal precision, PAEM reaches a better likelihood in 3 cases against 2 from the pure BW. Table 2 shows these cases. For all the 36 experiments, PAEM was better in 17 occurrences against 19 of the pure BW. However the difference in the vast majority of these cases lies in a nth decimal precision, which can be seen in Table 2, it represents a negligible probability.

4.2 Iterations As in the previous section, we started by checking our hypothesis through experiments using 50 different seeds to BW, excluding the best and the worst 5. From the 40 remaining we collected the best, the average, and the worst case concerning the BW initialization and its number of iterations to reach a convergence. As PAEM generates the parameters through a deterministic technique, it only needs one initialization. Figure 2 shows the average scenario. The other scenarios have a similar behavior.

Table 2: -Log-Likelihood comparison, cases where the difference exceeds a precision of one float point.

BW	PAEM	Δ%	
210.13	209.71	0.01%	favorable to PAEM
740.52	697.38	5.80%	favorable to PAEM
773.44	718.20	7.10%	favorable to PAEM
471.82	489.84	3.60%	favorable to BW
321.55	322.99	0.40%	favorable to BW

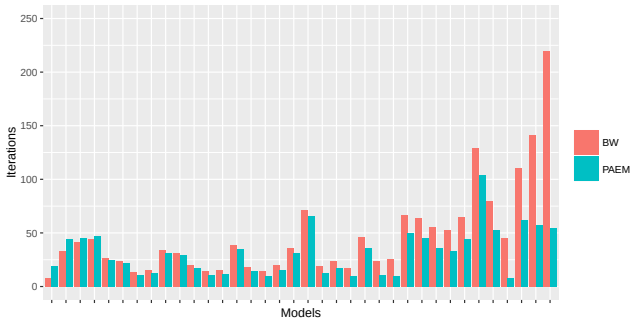

Figure 2: Average scenario for the required number of iterations to find a convergence. Experiments organized according to the models more favorable to BW (left) to the ones more favorable to PAEM (right).

In these pictures, we can clearly see PAEM requiring fewer iterations to find a convergence (right side), while just in a few cases, the random parameters outperformed PAEM (left side). Furthermore, these are retrieved from the experiments described in the Section 4.1, which shows an equivalent likelihood, between BW and PAEM, for 87% of the cases. Also, the far most significant scenario which PAEM performed poorly, loses with a difference of 3.6% (Table 2, BW=471.8).

Concerning all the results for the ordinary BW; 40 seeds for all the 12 series and the 2, 3, and 4 states model; the random values for BW got an average of 47.4 iterations against 25 from PAEM's. This shows a significant improvement for the initial parameters quality against the traditional random approach. Furthermore, in the vast majority of the tests, PAEM found a convergence with fewer iterations (Figure 2.

4.3 User Time Since both, BW and PAEM, tend to converge to the same likelihood and the cost to randomize values to BW is trivial, the real advantage of PAEM lies on a faster

convergence, which is given by fewer iterations derived by a better likelihood at the first iteration.

We performed time measurements to see how fast each procedure is in relation with BW. Although the running time is highly correlated with the number of EM iterations, a lower running time is the final goal, therefore, a more precise measure regarding the time actually used by BW and PAEM. In a standard machine, Intel i5, 2.3GHz, 8GB, a four states model had an average time of 0.34 seconds running with PAEM and 1.17 seconds running with BW. This difference is directly linked with the number of iterations. As described in the previous sections, in most cases, PAEM's pre-fitting tends to avoid at least one iteration of the forward-backward procedure, which costs $\vartheta(N^2T)$, which is more than PAA $\vartheta(T)$.

The user time spent, from both, had a strong correlation with their number of iterations. Specifically, BW had a correlation average of 93% and PAEM 76%. Which can be explained by the different seeds in BW and the lack of a precise control considering an ordinary machine running other applications. Also, PAEM's pre-fitting has a fixed running time for series with the same length, which has a different impact according to the series number of iterations necessary to find a convergence.

Now, considering the average scenario, we look for each of the 36 experiments. Thus, the radar showed by Figure 3, illustrates the total time spent in relation to the average scenario for each time series. From this figure, we can clearly see the time percentage difference from each technique and for each time series. This plot shows the overall better performance of PAEM, failing in just 6 cases, which are the time series 1, 10, 16, 21, 24, and 26. However, a difference in the case 26 is meaningless since the difference is 0.0004 in favor of BW.

Among these time series, a critical poor performance was achieved on the time series #1 and #10, which happens to be the shortest time series in the experiment. Considering BW and PAEM, respectively, for the first time series, considering an average case, it required 0.026 and 0.110 seconds. Time series X10, required, respectively, 0.025 and 0.064 seconds.

Considering larger models, we can verify that PAEM performed better, in average, for any number of states less or equal to 22. Further tests are required for larger models. Finally, we emphasize that, our code was extended from [31] and they do not have focus on performance. Therefore, the user time is far from optimal and the difference might be much less than the observed in our experiments.

5 Discussion

Through exhaustive tests, with different series, we found PAEM to be faster than BW with its traditional stochastic initialization. Its performance is due to, usually, fewer iterations in the EM procedure. In fact, as shown in Figure 2, in the vast majority of executions, the convergence is achieved using a fewer number of iterations than the pure BW. Furthermore, the time needed to the piecewise approximation is far smaller than one EM iteration.

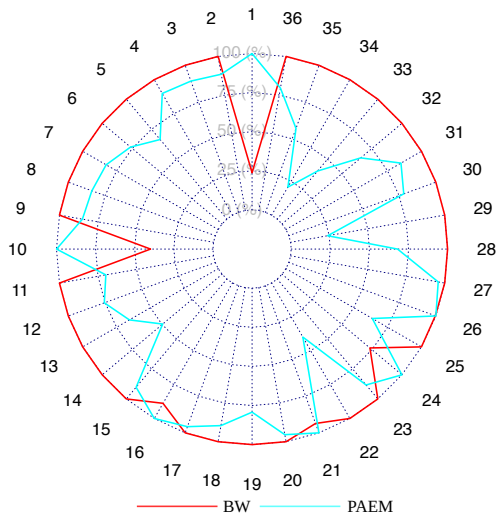

Figure 3: Radar chart showing the slowest process taking 100% of the time. An average scenario considering the user time.

However, it is important to emphasize the overall better performance and that PAEM does not aim to be an optimal solution. We focus on a simple alternative to the initial and usual randomization of parameters. Although there are other techniques that improve the original BW, PAEM lies in a simple initialization that is fast and easy to implement, making it a suitable alternative to performing HMM fitting. In fact, there are cases where authors related a faster solution using simpler MLEs [2]. Other techniques such as the Levenberg-Marquardt algorithm, can be used to maximize directly the likelihood, which can be faster than EM approaches [28].

For the future improvements, we shall focus on measuring the initial likelihood and the user time according to different kinds of data and distributions. PAEM is based on a simple Piecewise Aggregation technique. It may have a better global performance if a more advanced technique is used instead of Piecewise Aggregation. For this reason, we do not focus on measure the impact of w in its parameters. In a future work, we can focus on comparisons, such as the impact of different values of w and more robust techniques, like SAX [22] and its derivations. However, the time spent to pre-process the data must be lower than the original one. Otherwise, the overall performance might decrease. Another important test is to detect how efficient PAEM scales regarding models with a different number of states.

References

[1] R. AGRAWAL, C. FALOUTSOS, AND A. N. SWAMI, *Efficient similarity search in sequence databases*, in Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms, FODO '93, London, UK, UK, 1993, Springer-Verlag, pp. 69–84.

[2] R. M. ALTMAN AND A. J. PETKAU, *Application of hidden markov models to multiple sclerosis lesion count data*, Statistics in Medicine, 24 (2005), pp. 2335–2344.

[3] A. ARAPOSTATHIS AND S. I. MARCUS, *Analysis of an identification algorithm arising in the adaptive estimation of markov chains*, Mathematics of Control, Signals and Systems, 3 (1990), pp. 1–29.

[4] P. BALDI AND Y. CHAUVIN, *Smooth on-line learning algorithms for hidden markov models*, Neural Comput., 6 (1994), pp. 307–318.

[5] L. E. BAUM, T. PETRIE, G. SOULES, AND N. WEISS, *A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains*, The Annals of Mathematical Statistics, 41 (1970), pp. 164–171.

[6] C. BIERNACKI, G. CELEUX, AND G. GOVAERT, *Choosing starting values for the {EM} algorithm for getting the highest likelihood in multivariate gaussian mixture models*, Computational Statistics & Data Analysis, 41 (2003), pp. 561 – 575. Recent Developments in Mixture Model.

[7] O. CAPPE, V. BUCHOUX, AND E. MOULINES, *Quasi-newton method for maximum likelihood estimation of hidden markov models*, in Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, vol. 4, May 1998, pp. 2265–2268 vol.4.

[8] G. CELEUX, D. CHAUVEAU, AND J. DIEBOLT, *On Stochastic Versions of the EM Algorithm*, Research Report RR-2514, 1995.

[9] G. CELEUX AND G. GOVAERT, *A classification {EM} algorithm for clustering and two stochastic versions*, Computational Statistics & Data Analysis, 14 (1992), pp. 315 – 332.

[10] K.-P. CHAN AND A.-C. FU, *Efficient time series matching by wavelets*, in Data Engineering, 1999. Proceedings., 15th International Conference on, Mar 1999, pp. 126–133.

[11] I. B. COLLINGS, V. KRISHNAMURTHY, AND J. B. MOORE, *On-line identification of hidden markov models via recursive prediction error techniques*, IEEE Transactions on Signal Processing, 42 (1994), pp. 3535–3539.

[12] DATAMARKET!, *The open portal to thousands of datasets from leading global providers*. http://datamarket.com/, 2013.

[13] A. P. DEMPSTER, N. M. LAIRD, AND D. B. RUBIN, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society, series B, 39 (1977), pp. 1–38.

[14] R. DURBIN, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.

[15] J. FESSLER AND A. HERO, *Space-alternating generalized expectation-maximization algorithm*, Signal Processing, IEEE Transactions on, 42 (1994), pp. 2664–2677.

[16] G. FLOREZ-LARRAHONDO, S. BRIDGES, AND E. A. HANSEN, *Incremental estimation of discrete hidden markov models based on a new backward procedure*, in Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, AAAI'05, AAAI Press, 2005, pp. 758–763.

[17] A. GARG AND M. K. WARMUTH, *Inline updates for hmms.*, in Interspeech, ISCA, 2003.

[18] M. JAMSHIDIAN AND R. I. JENNRICH, *Conjugate gradient acceleration of the em algorithm*, Journal of the American Statistical Association, 88 (1993), pp. 221 – 228.

[19] F. KORN, H. V. JAGADISH, AND C. FALOUTSOS, *Efficiently supporting ad hoc queries in large datasets of time sequences*, in Proceedings of the 1997 ACM SIGMOD international conference on Management of data, SIGMOD 97, New York, NY, USA, 1997, ACM, pp. 289–300.

[20] V. KRISHNAMURTHY AND J. B. MOORE, *On-line estimation of hidden markov model parameters based on the kullback-leibler information measure*, IEEE Transactions on Signal Processing, 41 (1993), pp. 2557–2573.

[21] F. LEGLAND AND L. MEVEL, *Recursive identification of hmms with observations in a finite set*, in Decision and Control, 1995., Proceedings of the 34th IEEE Conference on, vol. 1, Dec 1995, pp. 216–221 vol.1.

[22] J. LIN, E. KEOGH, S. LONARDI, AND B. CHIU, *A symbolic representation of time series, with implications for streaming algorithms*, in Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, DMKD '03, New York, NY, USA, 2003, ACM, pp. 2–11.

[23] C. LIU AND D. B. RUBIN, *The ecme algorithm: A simple extension of em and ecm with faster monotone convergence*, Biometrika, 81 (1994), pp. pp. 633–648.

[24] C. LIU, D. B. RUBIN, AND Y. N. WU, *Parameter expansion to accelerate em: The px-em algorithm*, Biometrika, 85 (1998), pp. pp. 755–770.

[25] R. SALAKHUTDINOV, S. ROWEIS, AND Z. GHAHRAMANI, *Expectation-Conjugate Gradient: An Alternative to EM*.

[26] S. SIVAPRAKASAM AND K. S. SHANMUGAN, *A forward-only recursion based hmm for modeling burst errors in digital channels*, in Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE, vol. 2, Nov 1995, pp. 1054–1058 vol.2.

[27] M. A. TANNER AND W. H. WONG, *The calculation of posterior distributions by data augmentation*, Journal of the American Statistical Association, 82 (1987), pp. pp. 528–540.

[28] R. TURNER, *Direct maximization of the likelihood of a hidden markov model*, Computational Statistics & Data Analysis, 52 (2008), pp. 4147 – 4160.

[29] D. A. VAN DYK, X.-L. MENG, AND D. B. RUBIN, *Maximum likelihood estimation via the ecm algorithm: computing the asymptotic variance*, Statistica Sinica, (1995), pp. 55–75.

[30] G. C. G. WEI AND M. A. TANNER, *A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms*, Journal of the American Statistical Association, 85 (1990), pp. pp. 699–704.

[31] W. ZUCCHINI AND I. MACDONALD, *Hidden Markov Models for Time Series: An Introduction Using R*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis, 2009.