Genetic algorithm for burst detection and activity tracking in event streams*

Lourdes Araujo[1], José A. Cuesta[2] and Juan J. Merelo[3]

[1]Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Spain, lurdes@sip.ucm.es
[2]Grupo Interdisciplinar de Sistemas Complejos (GISC), Departamento de Matemáticas, Universidad Carlos III de Madrid, Spain, cuesta@math.uc3m.es
[3]Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada, Spain, jmerelo@geneura.ugr.es

Abstract. We introduce a new model for detection and tracking of bursts of events in a discrete temporal sequence, its only requirement being that the time scale of events is long enough to make a discrete time description meaningful. A model for the occurrence of events using with Poisson distributions is proposed, which, applying Bayesian inference transforms into the well-known Potts model of Statistical Physics, with Potts variables equal to the Poisson parameters (frequencies of events). The problem then is to find the configuration that minimizes the Potts energy, what is achieved by applying an evolutionary algorithm specially designed to incorporate the heuristics of the model. We use it to analyze data streams of very different nature, such as seismic events and weblog comments that mention a particular word. Results are compared to those of a standard dynamic programming algorithm (Viterbi) which finds the exact solution to this minimization problem. We find that, whenever both methods reach a solution, they are very similar, but the evolutionary algorithm outperforms Viterbi's algorithm in running time by several orders of magnitude, yielding a good solution even in cases where Viterbi takes months to complete the search.

1 Introduction and state of the art

Suppose you are the marketing manager of a publishing house which has recently released a book targeted at being a new best-seller, you have launched a marketing and PR campaign and want to know its impact. One thing you can do is to collect e-mails from public discussion forums on books and check for those that talk about the topic the new book deals with. Once these messages have been selected, you end up with a temporal sequence of events. The campaign can

* Supported by projects TIC2003-09481-C04 (L.A. and J.J.M.) and BFM2003-0180 (J.A.C.) from the Ministerio de Ciencia y Tecnología (Spain), as well as the project S-0505/ESP/000299 (J.A.C.) from the Comunidad de Madrid (Spain) and the joint project UC3M-FI-05-007 (J.A.C.) from the Comunidad de Madrid and Universidad Carlos III de Madrid (Spain).

be considered successful if after the targeted advertising campaign, references to the book show up as a burst of activity in your temporal series.

In the area of "topic detection and tracking" (TDT) [1], once the document topics have been identified, the sequence of documents for a particular topic can be regarded and analyzed as any other temporal series. There are many natural and social phenomena that produce such temporal series of events: seisms, books or CDs sales, news, e-mails and citations to a scientific paper, to name a few. In all of them, events roughly concentrate in bursts in which, loosely speaking, the frequency of their occurrence first rises, stays there for a while and then fades away. In a graphical representation of such sequences these bursts are more or less visible; however, a precise automatic detection of these bursts and of the frequency of events in them is not trivial because the sequence of events is a stochastic process and noise hides the relevant information. Even in the middle of the burst, events can apart from each other. On the contrary, even if there is no such burst, a few events may be close together. Discriminating whether these are just noise or a significant part of a burst is the real problem that we address in this paper.

Different statistical techniques[6, 3, 5] have been applied to analyse temporal changes in document streams. In particular, Kleinberg proposed in [6] a probabilistic automaton to model the frequency of document arrival, e-mails with a given topic in his case. High activity episodes or bursts correspond to intervals of high frequency of arrival. The most probable sequence of frequencies follows from Bayesian inference through Viterbi's dynamic programming algorithm [4, 7]. A similar analysis using an evolutionary algorithm (EA) instead was recently proposed by some of us [2].

However, there is a limitation in Kleinberg model: events can happen at any time instant, which is true for a certain kind of them (for instance, the problem that motivated Kleinberg, which was e-mail classification). However, the timescale of some events is so long that this may be too much information for a proper modeling. For instance, low intensity earthquakes have a timescale of days; sales are registered with a timescale of days or even weeks, and e-mails on a given topic in a discussion forum also have a timescale of days (an accurate registering of time leads to some modeling issues as the difference between day and night, for instance). A typical sequence of such events consists of the number of occurrences per day, per week, per month or other adequate time period, which is a discrete temporal sequence that cannot be correctly modeled by Kleinberg's model. That is the main reason why we propose here a new automaton model for such a kind of sequences and apply it to several data streams of this type. The search for the most probable sequence of frequencies is made with both Viterbi's algorithm and an EA. When the complexity of the temporal sequence is high enough, the latter, which we present here, is shown to perform much better, even in cases in which Viterbi is infeasible.

The rest of the paper is organized as follows: section 2 describes the probabilistic model, section 3 introduces the EA to obtain the most probable sequence

of event frequencies, and section 4 shows the performance in applications to different kinds of data. Results are summarized in section 5.

2 Potts model for event streams

Suppose we have an event log along T time units (days, months, years...) that registers the number of events which occurred at all times $t = 0, 1, \ldots, T$. If events arrive at a constant average frequency, λ, independently of each other, the number of events at any given time, n, follows a Poisson distribution

$$P(n|\lambda) = e^{-\lambda}\frac{\lambda^n}{n!}, \qquad n = 0, 1, 2 \ldots \tag{1}$$

Frequencies may be different at different times, so assuming independence of events occurring at different times, the probability that we observe a certain stream of events $\{n_1, n_2, \ldots, n_T\}$ will be given by

$$P(n_1, n_2, \ldots, n_T|\lambda_1, \lambda_2, \ldots, \lambda_T) = \prod_{t=1}^{T} P(n_t|\lambda_t), \tag{2}$$

where λ_t denotes the event frequency at time t, and $P(n|\lambda)$ is given by (1).

In a typical experiment we have the stream $\{n_1, n_2, \ldots, n_T\}$ and what we want to estimate is the sequence of frequencies which these events have occurred with. Thus we apply Bayesian inference and express

$$\begin{aligned} P(\lambda_1, \ldots, \lambda_T|n_1, \ldots, n_T) &= \frac{P(n_1, \ldots, n_T|\lambda_1, \ldots, \lambda_T)P(\lambda_1, \ldots, \lambda_T)}{P(n_1, \ldots, n_T)} \\ &= \frac{\exp\left\{\sum_{t=1}^{T}(n_t \ln \lambda_t - \lambda_t)\right\} P(\lambda_1, \ldots, \lambda_T)}{P(n_1, \ldots, n_T)\prod_{t=1}^{T} n_t!}, \end{aligned} \tag{3}$$

where we have inserted (2) once 1 has been substituted into it.

Our problem is now to make a sensible choice of the *prior* $P(\lambda_1, \ldots, \lambda_T)$, given the situation we want to model. To begin with, we will assume that the sequence of frequencies is a Markov process, i.e. the frequency that the events a time t have arrived with only depends on the frequency of arrival at the previous time step. This is, of course, a simplification; if the process that models the frequencies has memory, its modeling is hopeless unless we have further information on it. On the other hand, most stochastic processes in real life are Markov processes, so this is a reasonable assumption. Therefore,

$$P(\lambda_1, \ldots, \lambda_T) = P(\lambda_1)P(\lambda_2|\lambda_1)\cdots P(\lambda_T|\lambda_{T-1}). \tag{4}$$

For practical purposes we will assume that frequencies can only take values from a discrete set Λ. Then we take $P(\lambda_1) = 1/E$, with E the number of frequencies in Λ, i.e. any frequency is considered as likely to be observed initially as any other. This reflects our lack of knowledge about the initial the state. For $P(\lambda'|\lambda)$

we make the hypothesis that if the process has frequency λ at time $t-1$, then it will tend to have the same frequency at time t; so with probability p, $\lambda' = \lambda$, and with probability $1-p$, $\lambda' \neq \lambda$ and will equiprobably take any other value of the frequency. Thus,

$$P(\lambda'|\lambda) = p\,\delta_{\lambda',\lambda} + \frac{1-p}{E-1}(1-\delta_{\lambda',\lambda}), \tag{5}$$

where $\delta_{\lambda',\lambda} = 1$ if $\lambda' = \lambda$ and 0 otherwise. A more convenient rewriting is

$$P(\lambda'|\lambda) = \frac{1-p}{E-1}(1-\delta_{\lambda',\lambda} + e^K\,\delta_{\lambda',\lambda}) = \frac{1-p}{E-1}\,e^{K\delta_{\lambda',\lambda}}, \tag{6}$$

where we have introduced the new parameter $K = \log[p(E-1)/(1-p)]$.

If we now introduce the prior (4), with these choices, into equation (3),

$$P(\lambda_1,\ldots,\lambda_T|n_1,\ldots,n_T) = \frac{\exp\left\{\sum_{t=1}^T (n_t \ln\lambda_t - \lambda_t) + \sum_{t=2}^T K\delta_{\lambda_t,\lambda_{t-1}}\right\}}{Z(K;n_1,\ldots,n_T)}, \tag{7}$$

where

$$Z(K;n_1,\ldots,n_T) = E\left(\frac{E-1}{1-p}\right)^{T-1} P(n_1,\ldots,n_T)\prod_{t=1}^T n_t! \tag{8}$$

is a normalizing factor and therefore can also be written

$$Z(K;n_1,\ldots,n_T) = \sum_{\{\lambda_t\in\Lambda\}_{t=1}^T} \exp\left\{\sum_{t=1}^T (n_t\ln\lambda_t - \lambda_t) + \sum_{t=2}^T K\delta_{\lambda_t,\lambda_{t-1}}\right\}. \tag{9}$$

Expressions (7) and (9) define the well-known *Potts model* of Statistical Physics [8] in a one-dimensional lattice, with λ_t the Potts variable at site t, K the coupling constant and $\varphi(\lambda_t) = n_t\ln\lambda_t - \lambda_t$, for fixed n_t, an external field acting on λ_t.

Once we have an expression for $P(\lambda_1,\ldots,\lambda_T|n_1,\ldots,n_T)$ we can obtain the desired estimation for the sequence of frequencies $\{\lambda_1,\ldots,\lambda_T\}$ as that which maximizes this probability. Since $Z(K;n_1,\ldots,n_T)$ is independent on the frequencies and the numerator of (7) is an exponential, maximizing this probability is equivalent to maximizing the argument of the exponential (i.e. minus the energy of the configuration in the Potts model), namely

$$f(\lambda_1,\ldots,\lambda_T) = \sum_{t=1}^T (n_t\ln\lambda_t - \lambda_t) + K\sum_{t=2}^T \delta_{\lambda_t,\lambda_{t-1}}, \tag{10}$$

which turns into a well-defined *fitness function* for an evolutionary algorithm. Please note that the two sums have competing effects on the frequencies: the first one gets maximized when every λ_t is as close as possible to n_t, while the second one reaches its maximum when all frequencies are equal. The "coupling" K is then a parameter that tunes the "stiffness" of the estimation, i.e. the larger

K, the more new events will be assumed to have arrived with the same frequency as the previous ones. Playing with K we can smooth out the intrinsic noise that the data unavoidably contain, and at the same time capture net differences in the frequencies.

3 Evolutionary algorithm for event frequency tracking

We propose an EA to perform the search of the sequence of event frequencies which maximizes (10). First of all, we need to estimate the model parameters. An estimate of the minimum, λ_{min}, and maximum, λ_{max}, frequencies can be $\lambda_{min} = (1/2)T^{-1}$, $\lambda_{max} = 2M$, with T the longest interval without events, and M the maximum number of events registered in a time unit. The value of E, i.e. the number of different frequencies (states of the automaton) considered is then given by $E = \lambda_{max}/\lambda_{min}$. Thus the possible frequencies are $\lambda_\alpha = \alpha\lambda_{min}$, $\alpha = 1, \cdots, E$.

The choice of p (see 5) is rather arbitrary. However, it enters the model through K (see 6), and this parameter is rather insensitive to the precise value of p provided it is in the range ~ 0.3–0.7. Thus, after checking that other choices lead to the same results, we have taken $p = 0.5$.

The fitness function is directly provided by (10). In what follows we define the remaining ingredients of the EA.

3.1 Individuals and initial population

The most immediate representation of the individuals of our EA would be a sequence of T frequencies, one for each time unit. Accordingly, an individual would be a list of T genes g_t, where $g_t \in \{0, \cdots, E\}$ is the frequency at time t, α_t.

However, many adjacent times can be assigned the same frequency, so the sequence of transitions can be compacted. Thus, an individual is a variable length list, in which each position, or gene, represents a time interval with the same frequency. Each gene is composed of a frequency and of an identifier of the first and the last time of the interval.

g_1	g_2	·····	g_f
$\alpha_1, [1, t_2 - 1]$	$\alpha_{t_2}, [t_2, t_3 - 1]$	·····	$\alpha_{t_f}, [t_f, T]$

The initial population of our algorithm is composed of individuals composed of randomly generated sequences of frequency transitions. The simplest way of creating one such sequence is to choose a few times at random and use them to split the whole period of time into intervals, every one of which is assigned a random frequency. Some preliminary experiments have shown, however, that such a naive strategy gives rise to a search space much too large for the algorithm to be efficient. Accordingly, we propose a different strategy. We again choose a random set of times for splitting, but remove those for which the number of

events in the preceding interval differ in less than 50%. Afterwards, the first interval is assigned a random frequency and subsequent intervals are assigned a random higher frequency if they have more events than the preceding interval or a random lower frequency if they have less events.

3.2 Crossover operator

We have adopted the classic one point crossover, which creates two offsprings by splitting two individuals at a crossover point and swapping their second bits. Then, the best offspring replaces the worst parent (steady state, elitist strategy).

There are some details that have to be dealt with, though. The crossover point lies in genes g and g' of both parents, respectively. Thus after swapping, unless both g and g' have the same frequency, each of these genes will become two, one on the left of the crossover point and one on the right, with different frequencies. Several strategies have been tested, but the most efficient one is to leave them as two genes if the number of events at the crossover instant and the preceding instant differ more than 50%; otherwise both the left and right genes are assigned the frequency of g or g' at random, and thus converted back in a single gene.

3.3 Mutation operator

We have implemented three different mutation operators and each time a mutation occurs one of them is applied at random. The operators are:

1. Choose a gene at random and with equal probability increment or decrement its frequency to the next or previous one.
2. Join two consecutive genes to produce a single one with a frequency randomly taken from one of the original genes.
3. Split a gene in two and assign a different frequency to each piece: one of them is given the frequency of the original gene and the other one is incremented or decremented (depending on whether the number of events is larger or smaller than that of the other piece) a random amount. This operator is only applied if the resulting number of events at both sides of the partition differ more than 50%.

4 Experimental results

The present model relies on two assumptions: (i) that events occur with a Poisson distribution and (ii) that frequency changes occur with a constant probability and contiguous frequencies are uncorrelated. In order to test the Bayesian reconstruction with our EA independently of these two assumptions we have first tested the model against data streams artificially created using Poisson distributions of different frequencies. The sequence `art_poisson1` presents short periods of constant frequency and small frequency jumps, and the sequence

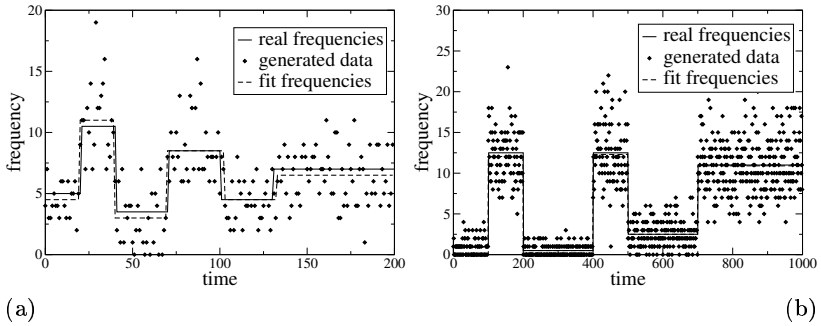


Fig. 1. Artificial sequences created to test the EA: `art_poisson1` (a), and `art_poisson2` (b). We plot the exact sequence of frequencies (full lines), the number of events generated with this sequence along 200 (a) or 1000 (b) time steps (dots), and the result from the EA as well as Viterbi's algorithm (dashed line; both results are indistinguishable in the plot).

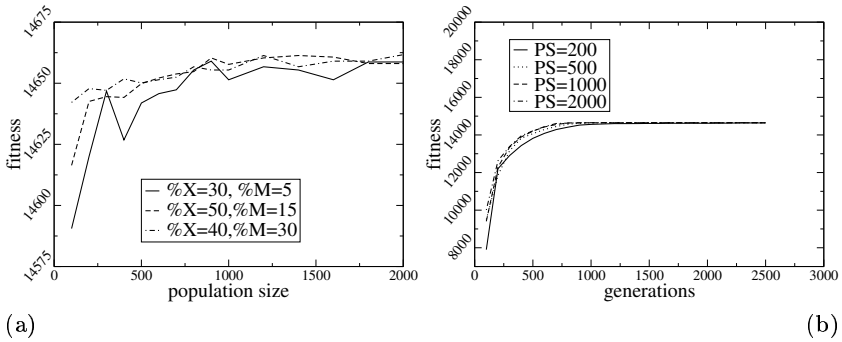


Fig. 2. (a) Final fitness attained with different population sizes for several values of crossover and mutation rates. (b) Evolution of the fitness along 10^4 generations (crossover rate = 50%, mutation rate = 15%), for different population sizes.

`art_poisson2` presents long periods of constant frequency and large frequency jumps. The outcome of our EA has been compared with that of Viterbi's algorithm (an exhaustive search algorithm for Markov chains) as well as with the exact sequence of frequencies. Both algorithms have been implemented in C++ and run on a Pentium IV 2.4MHz and 1Gb of memory running Linux. Results appear in Figure 1. First of all, we can observe that Viterbi's algorithm reproduces with high accuracy the sequence of frequencies, which proves the validity of the Bayesian inference applied to this model, but the EA yields results which are indistinguishable from them, which proves the validity of the EA —at least for these simple sequences. The next step taken has consisted in tuning the parameters of the EA. For this purpose we have chosen `art_poisson2`. Figure 2 shows the final fitness attained by the EA for different population sizes (a) as well as its evolution with time for a fixed population (b). Plotted data are aver-

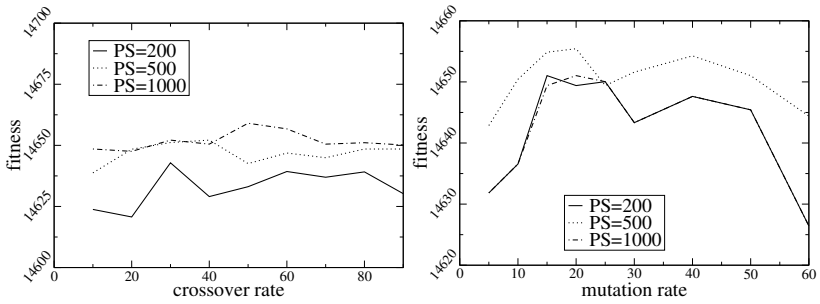


Fig. 3. Fitness reached with different values of the crossover rate (with a 15% of mutation) (a) as well as of the mutation rate (with a 50% of crossover) (b), for several population sizes. Total number of generations: 10^4.

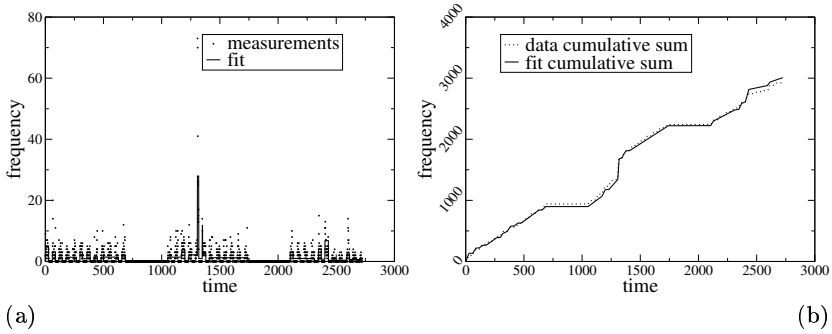


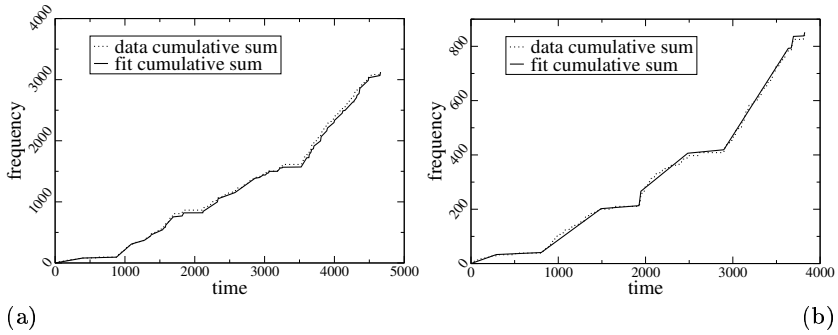
Fig. 4. Fit to the daily number of earthquakes of magnitude ≥ 2 which occurred in Spain in the period 2002/01/01–2004/11/22.

ages over 5 different EA runs. Figure 2(a) shows that fitness improves with the population size for any setting of the remaining parameters, although beyond 10^3 individuals and using intermediate values for crossover and mutation rates, no further improvement is obtained. Figure 2(b) shows the fast increase of fitness to its maximum, which is faster the larger the population is (although actual differences are negligible). Figure 3 shows the effect of crossover and mutation on the fitness, also for `art_poisson2`. It can be observed that the best results are obtained with a crossover rate $\sim$ 30–60% and a mutation rate $\sim$ 15–25%.

4.1 Results on real data

We have applied our EA to real streams of events of very different nature. The first one, shown in Figure 4, provides the daily number of earthquakes of magnitude ≥ 2 which occurred in Spain in the period 2002/01/01–2004/11/22[1].

[1] Data taken from the "Advance National Seismic System Catalog", web page www.ncedc.org/cnss/catalog-search.html.



(a) (b)

Fig. 5. Time sequence of comments on 'blog' and 'Google' during the period January 2002-January 2006.

Figure 4(a) illustrates the fit produced by the EA with 10^3 individuals, a 30% of crossover, a 5% of mutation and run for 5×10^3 generations. Figure 4(b) is a cumulative plot of the same results. The goodness of the fit is more evident in this latter plot, so the remaining data are plotted using this representation. The second and third set of data are formed by the comments on 'blog' and 'Google', respectively, sent to all blogs hosted in Blogalia (http://blogalia.com) during the period January 2002-January 2006. A cumulative plot of these data, as well as the fits produced by the EA with 10^3 individuals, a 50% of crossover, a 15% of mutation and run for 10^4 generations, appear in Figure 5. Despite being data of a very different nature, the fit is as good as that for the earthquakes. One of the most important aspects to remark about our experiments is the comparison of the execution times needed by the EA and by the Viterbi's algorithm. As it can be seen in Table 1, Viterbi required about two orders of magnitude more time to reach the solution than the EA. In fact, in two of the cases Viterbi was stopped without reaching a solution, and the time to get it was estimated extrapolating from the time required to compute every time step. The main reason for such an impressive improvement in performance is the fact that the EA conducts a search very much guided by the heuristics on the particular problem under study that can be implemented in the evolution operators (in our case, for instance, the way crossover and mutation are implemented eliminates trials which assign different frequencies to segments with similar number of events). This dramatically reduces the size of the search space, so much as to render feasible problems that

	Viterbi		EA	
earthquakes	7140862 s	(> 82 days)	27514.2 s	(7.64 hours)
coment_blogs	697412 s	(> 8 days)	25139.3 s	(7.00 hours)
coment_google	237800 s	(> 2 days)	35609.4 s	(9.89 hours)

Table 1. Execution times required by the EA and estimated for Viterbi.

would not be so with a standard algorithm like Viterbi's. These two elements: the accuracy of the results, and the dramatic increase in performance, justify the application of an EA to this problem.

5 Conclusions

In this paper we have proposed a model for detection and tracking of bursts in data streams coming from a wide range of problems. The model applies to those problems which are well represented by a discrete temporal series where we have a log of the number of events every time unit (day, week, month, year...). We model event occurrences by Poisson distributions and apply Bayesian inference to find the sequence of Poisson parameters that maximizes the likelihood. The problem is shown to be equivalent to minimizing the energy of the well-known Potts model of Statistical Physics. We use the negative of this energy as the fitness of a special purpose evolutionary algorithm to solve this problem, and apply it to streams of data obtained from earthquake detection and from weblog comments. The results are very similar to those obtained from Viterbi's algorithm, which is guaranteed to find the absolute maximum of such problems. However, execution times for the evolutionary algorithm are about two orders of magnitude smaller than those employed by Viterbi, thus leaving the evolutionary algorithm as the only available tool to reach a solution in a reasonable time for real collections of data.

References

1. James Allan. *Topic Detection and Tracking: Event-Based Information Organization.* Kluwer Academic Publishers, 2002.
2. Lourdes Araujo and Juan J. Merelo. Automatic detection of trends in dynamical text: An evolutionary approach, 2006. Available from http://arxiv.org/abs/cs.IR/0601047, OAI: arXiv.org:cs/0601047.
3. Ella Bingham, Ata Kabán, and Mark Girolami. Topic identification in dynamical text by complexity pursuit. *Neural Process. Lett.*, 17(1):69–83, 2003.
4. G. D. Forney. The Viterbi algorithm. *Proceedings of The IEEE*, 61(3):268–278, 1973.
5. Mark Girolami and Ata Kaban. Simplicial mixtures of Markov chains: Distributed modelling of dynamic user profiles. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16.* MIT Press, Cambridge, MA, 2004.
6. J. Kleinberg. Bursty and hierarchical structure in streams. In *Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 91–101. ACM, 2002.
7. Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., 1990.
8. F. Y. Wu. The Potts model. *Review of Modern Physics*, 54:235–268, 1982.