

# Recurrent Boosting for Classification of Natural and Synthetic Time-Series Data

Robert D. Vincent<sup>1</sup>, Joelle Pineau<sup>1</sup>, Philip de Guzman<sup>2</sup>, and Massimo Avoli<sup>2</sup>

<sup>1</sup> School of Computer Science, McGill University, Montreal, Quebec, Canada  
{bert, jpineau}@cs.mcgill.ca

<sup>2</sup> Montreal Neurological Institute, McGill University, Montreal, Quebec, Canada  
philip.deguzman@mail.mcgill.ca, massimo.avoli@mcgill.ca

**Abstract.** Boosted ensemble classifiers have a demonstrated ability to discover regularities in large, poorly modeled datasets. In this paper we present an application of multi-hypothesis AdaBoost to detect epileptiform activity from electrophysiological recordings. While existing boosting methods do not account automatically for the sequence information that is available when analyzing time-series data, we present a recurrent extension to AdaBoost, and show that it improves classification accuracy in our application domain.

Medical treatment design has long been the exclusive domain of clinical experts. However, recently there has been a growing interest in automatically optimizing *adaptive treatment strategies* for the management of chronic diseases. The challenge is in developing sequences of treatments which adapt to a patient's characteristics and the disease's progression [1]. Additionally, there has been much recent interest in using automatic techniques to classify neurological time-series data [2]. There are tremendous opportunities in applying automated learning and discovery techniques to these classes of problems.

The optimization of an adaptive treatment strategy can be cast as a reinforcement learning problem [1]. Reinforcement learning addresses the problem of optimizing action sequences in dynamic and stochastic systems [3]. In this paradigm, the state of the system represents the patient's medical history, and the goal is to use direct experimentation with the system to learn, for each state, the optimal treatment strategy (or *policy*). Reinforcement learning unfortunately tends to require large amounts of data to reach an optimal strategy. This is impractical where data is sparse and expensive, as is often the case with human medicine. The best way to reduce data requirements is to impose strong constraints on the state representation.<sup>1</sup> Thus a significant challenge is finding a good compact state representation for a patient's medical history.

In this paper we focus on the problem of learning a compact state representation for epileptic events. Epilepsy is a brain disorder characterized by seizures (also known as *ictal* events) resulting from episodes of abnormal electrical activity in the brain. It affects about 1% of the population [4], of which at least 25% do

---

<sup>1</sup> A secondary technique is to impose strong constraints on the policy space, but this generally requires a known state representation.

not respond to anti-epileptic medication [5]. For these non-responsive patients, treatment by electrical stimulation has recently emerged as a promising alternative therapy [6]. The technology is relatively simple: a small pacemaker-like device that applies mild electrical stimulation to the nervous system is implanted in the patient. The optimization of an adaptive treatment strategy for such a device requires a compact state representation, as it is likely that limited amounts of data will be available for learning. Therefore we seek methods for classifying epileptic states from electrical field potential recordings.

In this paper we attempt to detect epileptic states by performing classification problem over fixed time frames, and we investigate the use of boosting techniques to discover information about key features for our state representation. Though this is not always well recognized, ensemble methods such as AdaBoost provide a principled and efficient mechanism for feature selection in large, poorly modeled datasets [7,8]. However, existing boosting methods do not naturally account for the sequential nature of time-series data, such as electrophysiological recordings. We present a new recurrent formulation of AdaBoost, in which the classification of prior time frames is included in the feature vector of the current time frame. This technique distinctly improves classification accuracy in our application, especially the detection of rare events. We also evaluate the performance of recurrent AdaBoost using a synthetic dataset from the UCI database [9] and demonstrate improved classification accuracy compared to standard AdaBoost. While we do not provide a formal analysis of the properties of boosting under the recurrent formulation, this will be an interesting line of future research.

## 1 Problem Description

Epileptiform signals can be separated into long *normal* phases, with periodic *ictal* events that may span several minutes. They are also characterized by brief *interictal* events, sometimes called *spikes*.

The problem of automated real-time detection and prediction of epileptic seizures using electrophysiological recordings has been investigated extensively, yielding a variety of approaches, including neural networks [10], wavelet methods [11], and nonlinear time series analysis [12]. However these results are not sufficiently interpretable to build compact state representations.

### 1.1 Data Recordings

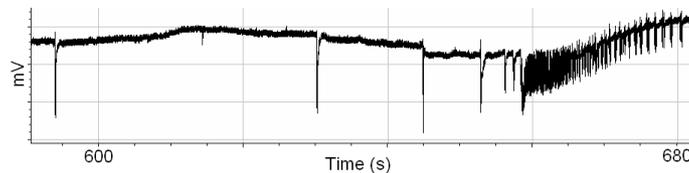
The data used in this study are field potential recordings of seizure-like activity in slices obtained from rat brains [13]. The recordings were made using micro-electrodes inserted in the regions of interest and sampled at a rate of 5012.5 Hz. The recordings were filtered to roll off frequencies above 100 Hz. This study used three separate brain slices. In each slice, neural activity was recorded in three different channels placed in different brain structures, thus yielding a total of nine data traces. These recordings are between 10.5 and 13 minutes in length.

## 1.2 Signal Processing

Each data trace was processed as a series of nonoverlapping frames consisting of 4096 samples (0.82 sec). Each frame was normalized by subtracting the mean and dividing by the full range of the entire frame. The per-frame mean, range, and energy (the sum of squared deviations from the mean) were saved for use as features in the classification. Each frame was then apodized with a Hann window and converted to a power spectrum using the discrete fast Fourier transform. Because the signals were low-pass filtered at 100Hz, only the first 80 frequency bands were used as features, representing a frequency range of approximately 1-98 Hz. The real and imaginary components of each band of the FFT were combined into a single magnitude, giving 83 features per frame (the frequency bands, plus mean, range and energy). Each trace yielded between 731 and 947 usable frames, for an overall total of 7692 frames.

## 1.3 Labeling

Each of the channels of the recordings was segmented into *normal*, *spike*, or *ictal* (or seizure) periods based on guidance from an expert. This classification was somewhat qualitative and performed by visual analysis. As can be seen in Fig. 1, the events are reasonably distinctive. Spikes were noted only for the duration of the most prominent portion of the spike waveform, giving a typical spike length of 50 milliseconds. The majority (82%) of the frames was classified as normal, with about 3% classified as an interictal spike and 14% classified as ictal. We have also made this labeled dataset publicly available [14].



**Fig. 1.** An example recording, showing several spikes and an ictal event (far right)

## 2 Algorithmic Approach

Boosting is a general supervised learning technique that seeks to combine an ensemble of simple, easily chosen classification rules (or *hypotheses*) into a single strong hypothesis. Most boosting algorithms proceed in a series of rounds in which a new weak hypothesis is trained according to a labeled set of training examples. After each round, the distribution of the training examples is updated to increase the weights of those examples that were improperly classified in the current round. The final strong hypothesis is formed by a weighted combination of the weak hypotheses [15].

## 2.1 AdaBoost

The general boosting framework specifies neither how distributions and weights are updated, nor how the weak hypotheses are to be combined. The AdaBoost (“adaptive boosting”) algorithm was invented by Freund and Schapire [7]. Our work uses AdaBoost.MH (illustrated as Algorithm 1), which is a multiclass extension of AdaBoost [16] that generalizes both the distributions and the weak learners over a set of possible labels. We specifically use “real” AdaBoost.MH, which outputs a real-valued confidence prediction for each class.

Our choice of AdaBoost was motivated primarily by the relative simplicity of the final classifier. While perhaps less amenable to human interpretation than a decision tree, a boosted classifier can yield insights into the structure of a poorly characterized problem by weighting features according to their discriminative power [8]. Also, while the algorithm’s performance is influenced by the choice of weak learners, the final strong hypothesis can often be evaluated very efficiently.

The use of the AdaBoost family of algorithms was also influenced by recent work in music genre classification which revealed AdaBoost as a powerful classification approach for complex time-series signals [17].

We use the freely available AdaBoost.MH implementation BoosTexter 2.1 [18], which includes weak learners consisting of simple decision stumps over continuous attributes. While this implementation was intended for text processing applications, it is general enough for our application.

We use the features described in Sect. 1.2 to form the feature domain  $X$ .

---

### Algorithm 1. Discrete AdaBoost.MH [16]

---

Given:  $(x_1, Y_1), \dots, (x_m, Y_m)$  where  $x_i \in \mathcal{X}$ ,  $Y_i \subseteq \mathcal{Y}$

Initialize  $D_1(i, \ell) = 1/(mk)$

**for**  $t = 1, \dots, T$  **do**

    Train weak learner using distribution  $D_t$

    Get weak hypothesis  $h_t : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$

    Choose:

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 + r_t}{1 - r_t} \right), r_t = \sum_{i, \ell} D_t(i, \ell) Y_i[\ell] h_t(i, \ell)$$

    Update:

$$D_{t+1}(i, \ell) = \frac{D_t(i, \ell) \exp(-\alpha_t Y_i[\ell] h_t(x_i, \ell))}{Z_t}$$

    (Where  $Z_t$  is a normalizing constant chosen such that  $\sum_i D_{t+1}(i) = 1$ )

**end for**

Output final hypothesis:

$$H(x, \ell) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x_i, \ell) \right)$$


---

## 2.2 Recurrent AdaBoost

AdaBoost does not directly represent any dependencies between events; each training example is assumed to be drawn independently randomly from the set  $X$ . For time series data it is likely that the classification of prior frames in the series will provide useful information for the classification of later frames.

The most obvious way to test this is to incorporate features from prior time frames  $x_{i-1}, \dots, x_{i-N}$  with features of the current time frame  $x_i$ . This method, which we call *AdaBoost with Memory*, is conceptually simple and maintains the good theoretical properties of boosting. However, it scales badly for domains with a large feature space.

Instead, we propose to use the *classification labels* of prior time frames. We train a classifier  $f$  such that  $y_i = f(x_i, y_{i-1}, \dots, y_{i-N})$ , where  $x_i$  is the input feature of frame  $i$ ,  $N$  is the number of prior predictions considered, and  $y_i$  is the set of real numbers corresponding to the class membership scores output by AdaBoost.MH. We call this algorithm *Recurrent AdaBoost*. It scales nicely with history size, assuming a small number of classes (3 in our case). A problem with  $K$  classes and  $N$  recurrent time steps adds  $NK$  features to the input vector.

Our recurrent approach requires inserting two steps in the AdaBoost training procedure. First, during initialization we set all of the prior labels in our training examples to zero. Second, these labels must be updated at the end of each round of training. The testing procedure also must be modified slightly in cases where test frames are processed in a batch manner. It is necessary to iterate classification of the test set (up to  $N$  times) to allow full incorporation of the classifier information. This is not necessary when test examples are presented in an order consistent with the time-series.

## 3 Experimental Evaluation

### 3.1 Method

In this section, we investigate the performance of boosting for the classification of epileptic brain activity from electrophysiological signals. We consider three different classification approaches:

$$\begin{aligned} y_i &= f(x_i) && \text{Standard AdaBoost} \\ y_i &= f(x_i, x_{i-1}, \dots, x_{i-N}) && \text{AdaBoost with Memory} \\ y_i &= f(x_i, y_{i-1}, \dots, y_{i-N}) && \text{Recurrent AdaBoost} \end{aligned}$$

In the control experiment, which we call *Standard AdaBoost*, each feature vector includes the 83 scalar values associated with the current time frame only.

In the second experiment, which we call *AdaBoost with Memory*, each feature vector includes the features of both the current time window and the prior time window for a total of 166 scalar values. This method can be extended to longer memory, but we did not try this because of the substantial training time required.

In the third experiment, which we call *Recurrent AdaBoost*, the input feature vector includes the 83 standard features with the addition of the output weights for each class, for each of  $N$  prior windows (where we vary  $N$  from 1 to 5.)

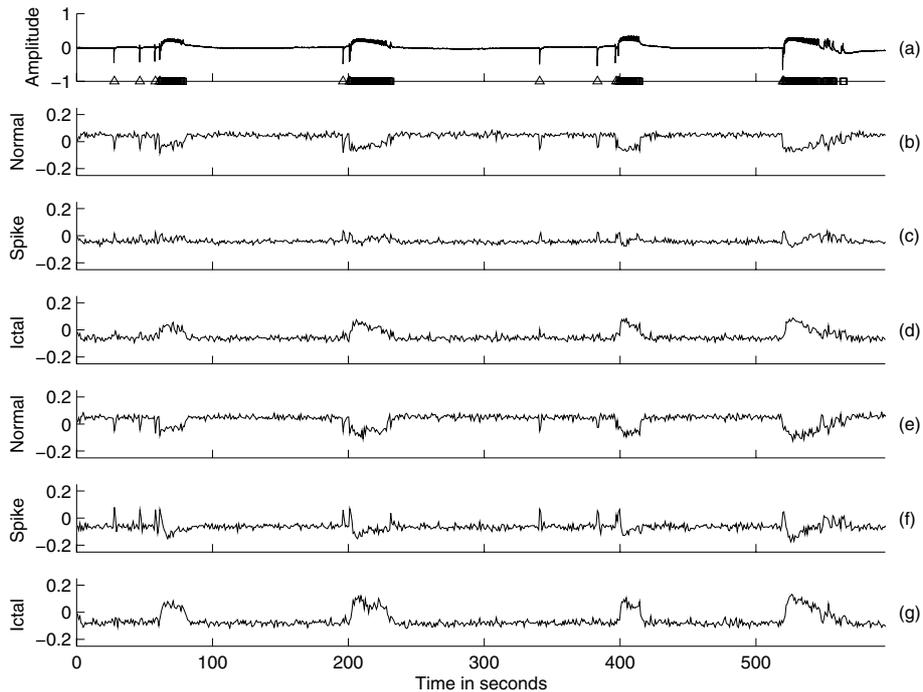
In each experiment, we performed three train/test folds using six traces as the training set and three traces as the test set. Training proceeded for 300 rounds, as the classification error leveled off after that point.

### 3.2 Results

We begin by considering an illustrative example. Figures 2b, 2c, and 2d show the classifier outputs for a representative test trace, using Standard AdaBoost. While overall results in this case were good (93% accuracy), only 10 of 12 spike frames (83%) and 82 of 119 ictal frames (69%) were correctly classified.

Figures 2e, 2f, and 2g show classifier outputs using Recurrent AdaBoost with the predictions of two prior frames. Here all 12 spike frames were properly identified, and the recognition of ictal frames increased to 102 out of 119 (86%).

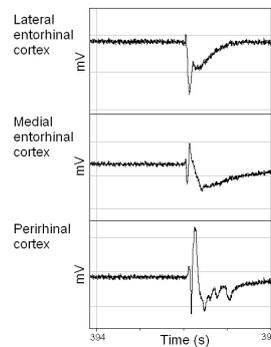
We now present a more formal comparison of the approaches. We achieved average overall accuracy greater than 90% with all methods considered.



**Fig. 2.** Results for classification of one channel. (a) The original trace. (b) (c) (d) The per-class confidence values using Standard AdaBoost. (e) (f) (g) The per-class confidence values using Recurrent AdaBoost with two prior time frames.

Results for all cases are summarized in Table 1. For Standard AdaBoost, the variance in accuracy among train/test folds was relatively high, ranging from 90% to 97%. Recognition of spikes was quite poor. Spike events may be especially difficult for our detector, because of both their short duration and their relatively rarity (3% of all frames). In some cases the classifier tended to classify spikes as ictal events. This may reflect variability in the spikes, which can resemble brief ictal events (see Fig. 3).

In the AdaBoost with Memory case, all features from the prior frame are concatenated with all features from the current frame. This approach shows a large improvement over Standard AdaBoost, and markedly reduced the variance in the accuracy. Note especially the improved detection of interictal spikes.



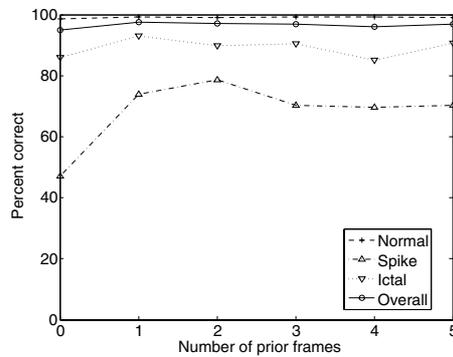
**Fig. 3.** Three channels recording a spike at approximately the same time. The bottom spike shows a long “ictal” tail.

**Table 1.** Summary of experimental results. Row labels reflect ground truth, column labels reflect classification results.

Experiment		Normal	Spike	Ictal	Total	Class%	Overall%	Range%
Standard AdaBoost	Normal	<b>6209</b>	18	67	6294	99	95	90–97
	Spike	35	<b>119</b>	99	253	47		
	Ictal	97	65	<b>983</b>	1145	86		
AdaBoost with Memory	Normal	<b>6242</b>	15	37	6294	99	97	93–99
	Spike	48	<b>187</b>	18	253	74		
	Ictal	92	15	<b>1038</b>	1145	91		
Recurrent AdaBoost (1 prior)	Normal	<b>6253</b>	16	25	6294	99	98	94–99
	Spike	49	<b>187</b>	17	253	74		
	Ictal	69	12	<b>1064</b>	1145	93		
Recurrent AdaBoost (2 prior)	Normal	<b>6239</b>	22	33	6294	99	97	92–99
	Spike	42	<b>199</b>	12	253	79		
	Ictal	101	15	<b>1029</b>	1145	90		
HMM	Normal					97	94	
	Spike					45		
	Ictal					78		

Results for Recurrent AdaBoost are shown for two cases, incorporating the predictions for either one or two prior frames. Incorporating one prior frame, there is a strong improvement over Standard AdaBoost in classifying both spikes and ictal events. Incorporating two prior frames provides no consistent benefit. These results are comparable to those of AdaBoost with Memory, but with less training time, given the smaller size of the feature space.

We evaluated Recurrent AdaBoost when incorporating predictions for 1–5 prior frames into the feature vector. These results are summarized in Fig. 4. There is little improvement beyond two frames, suggesting that, for our dataset, there is little added information in more distant time frames.



**Fig. 4.** Results for Recurrent AdaBoost using varying numbers of prior frames

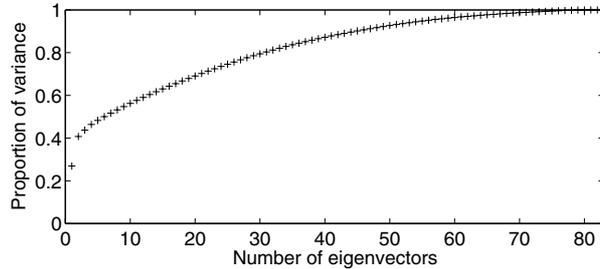
These results show solid detection of the various epileptic states. However, spike detection accuracy is at most 79%. In conversations with experts with years of experience reading recordings of the type shown in Fig. 1, the consensus is that given a full trace it is fairly easy to differentiate spiking and ictal events. However, given only data up to time  $t$ , it is difficult to predict whether a subsequent burst of energy is either a spike or the onset of an ictal event.

We also show results obtained using a standard time-series approach, the Hidden Markov Model (HMM) [19].<sup>2</sup> The results for classification of spike events are comparable to those of Standard AdaBoost but significantly worse than those of Recurrent AdaBoost. Classification of ictal events is worse than both Standard and Recurrent AdaBoost.

### 3.3 Feature Extraction

To better characterize the data, we performed principal components analysis of the 83 features that form our input space. Figure 5 shows that at least 50% of the principal components are required to reconstruct 90% of the variance

<sup>2</sup> Observation probabilities were modeled assuming each input feature follows a univariate Gaussian. All parameters were derived from the labeled training data.



**Fig. 5.** Principal components analysis of 83 features

in our data. We also observed much overlap between the spike class and the ictal/normal classes in principal components space. Both this analysis and the prior literature on the topic suggest that epileptic state detection from electrical signals is difficult, especially for spike events.

We also examined the strong hypotheses produced by AdaBoost.MH for all of the experiments. We observed a number of striking regularities.

In all recurrent examples, the first weak hypothesis recruited was either frequency band 62 or 63, corresponding to frequencies of 76 or 77 Hz. High values in these bands favor a normal classification, whereas low values weight towards ictal classification. Frequency bands 6–8 ( $\sim 7$ –10 Hz) were consistently recruited early. Low values in these bands favor normal classification, whereas high values favor ictal classification.

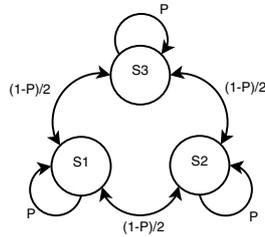
In most cases, energy was recruited in the first 20 rounds. A high energy value resulted in a strong weighting toward a spike classification. A similar effect was seen for the range feature.

In recurrent cases, prior labels primarily acted as a source of hysteresis in the system: prior labels of ictal or normal biased the present frame towards either ictal or normal, respectively.

### 3.4 Validation

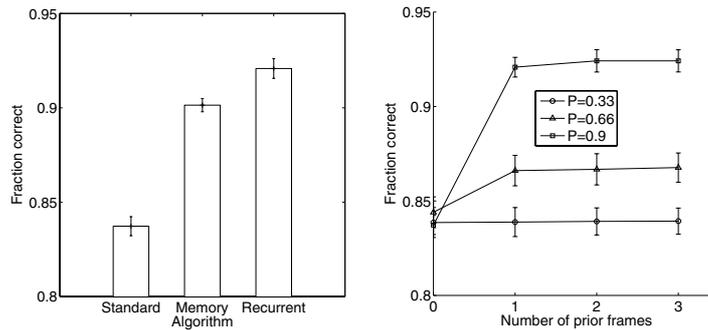
We also performed validation of Recurrent AdaBoost with synthetic data by adapting a similar task of known difficulty, the “waveform” classification problem from the UCI Machine Learning repository [20,9,21]. This task requires discrimination among three classes of 21 noisy continuous features. When the classes are chosen uniformly randomly in sequence (i.e. when the examples are generated i.i.d.), Bayes optimal performance is 86% [9]. We modified the problem by sampling examples from these three classes in a nonuniform sequence using the simple Markov model illustrated in Fig. 6 so as to produce time-series data.

We repeated each of the algorithms using 12 train/test rounds of 5000 training examples and 1000 testing examples, varying the transition probability  $P$ .



**Fig. 6.** Simple Markov model used to generate synthetic data. The parameter  $P$  sets the probability that the output class of an example is the same as that of the prior example. Initial states are chosen uniformly randomly.

The results are summarized in Fig. 7. The bar graph on the left shows the results for  $P = 0.9$  for Standard AdaBoost (left), AdaBoost with Memory (middle), and Recurrent AdaBoost using one prior prediction (right). The line graph on the right shows the results for Recurrent AdaBoost when  $P = 0.33$ ,  $P = 0.66$ , and  $P = 0.9$ , varying the number of prior predictions.



**Fig. 7.** Results for experiments with synthetic data. Error bars show 95% confidence intervals over 12 folds.

When no prior information is used, the results are similar to the Bayes optimal value (86%) for the uniformly random case; this is the Standard AdaBoost algorithm. However, when prior information is incorporated, we achieve significantly improved performance, as the algorithm is able to exploit the time-series information we added to the problem. Unsurprisingly, the improvement is largest when the value of  $P$  is highest, and only a single prior prediction is needed to capture the first-order Markov model. Recurrent AdaBoost performs slightly better than AdaBoost with Memory on this domain. We speculate that Recurrent AdaBoost achieves this by forming a smoothed summary of the state history.

## 4 Discussion

We propose a new way to apply boosting to time-series data by recurrent incorporation of class predictions into the feature vector. We show that this approach improves classification results in experiments with both real and synthetic data. We also contribute a new labeled dataset for time-series classification [14].

We also provide the first empirical evidence that AdaBoost can be used to characterize epileptic states in neurophysiological recordings. This task is difficult because of the large feature space, the unbalanced class distributions, the limited availability of training data, and the great variability of these recordings.

These findings show robust detection of key epileptic states. Recognition of interictal spikes was the most problematic, exhibiting high variance over the test cases. Note however that the training set is very small for this class, at most 204 examples for an 83-dimensional feature space. Furthermore, the class has strong overlap with others over the principal components of the feature space. In the future, we hope to investigate whether a similar approach may be used to classify subtler signals, such as those of cognitive states.

Our investigation was limited to using very simple weak learners. There is evidence that more sophisticated weak learners may yield a better strong hypothesis [17]. Other methods for applying boosting to time series data involved modifying the weak learners to account for time or spatial relationships [22,23]. This may be something to consider in the future.

We do not at this time provide a formal analysis of the convergence properties of Recurrent AdaBoost. The main challenge is the fact that the input set is not stationary due to its dependence on the classification of prior instances. This raises interesting theoretical questions which will be addressed in the future.

**Acknowledgments.** The authors gratefully acknowledge the financial support of the Canadian Institutes of Health Research (CIHR) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

1. Murphy, S.A.: An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine* **24** (2005) 1455–1481
2. Mitchell, T.M., Hutchinson, R., Niculescu, R.S., Pereira, F., Wang, X., Just, M., Newman, S.: Learning to decode cognitive states from brain images. *Machine Learning* **57** (2004) 145–175
3. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* **4** (1996) 237–285
4. Hauser, W.A., Hesdorffer, D.C.: *Epilepsy: Frequency, Causes and Consequences*. Demos, New York (1990)
5. Kwan, P., Brodie, M.J.: Early identification of refractory epilepsy. *New England Journal of Medicine* **342**(5) (2000) 314–319
6. Uthman, B.M., Reichl, A.M., Dean, J.C., Eisenschenk, S., Gilmore, R., Reid, S., Roper, S.N., Wilder, B.J.: Effectiveness of vagus nerve stimulation in epilepsy patients: a 12 year observation. *Neurology* **63** (2004) 1124–1126

7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55** (1997) 119–139
8. Viola, P., Jones, M.: Robust real-time object detection. *International Journal of Computer Vision* **57**(2) (2004)
9. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. *Machine Learning* **6** (1991) 37–66
10. Chiu, A.W.L., Daniel, S., Khosravani, H., Carlen, P.L., Bardakjian, B.L.: Prediction of seizure onset in an *in-vitro* hippocampal slice model of epilepsy using gaussian-based and wavelet-based artificial neural networks. *Annals of Biomedical Engineering* **33**(6) (2005) 798–810
11. Khan, Y.U., Gotman, J.: Wavelet based automatic seizure detection in intracerebral electroencephalogram. *Clinical Neurophysiology* **114** (2003) 898–908
12. Martinerie, J., Adam, C., Le Van Quyen, M., Baulac, M., Clemenceau, S., Renault, B., Varela, F.J.: Epileptic seizures can be anticipated by non-linear analysis. *Nature Medicine* **4**(10) (1998) 1173–1176
13. De Guzman, P., D’Antuono, M., Avoli, M.: Initiation of electrographic seizures by neuronal networks in entorhinal and perirhinal cortices *in vitro*. *Neuroscience* **123** (2004) 875–886
14. Vincent, R.D., Pineau, J., de Guzman, P., Avoli, M.: Labeled epileptiform data. <http://www.cs.mcgill.ca/~jpineau/datasets/epilepsy.tar.gz> (2006)
15. Schapire, R.E.: The boosting approach to machine learning: an overview. In Denison, D.D., Hansen, M.H., Holmes, C.C., Mallick, B., Yu, B., eds.: *Nonlinear Estimation and Classification*. Springer (2003) 149–172
16. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. *Machine Learning* **37** (1999) 297–336
17. Bergstra, J., Casagrande, N., Eck, D.: Two algorithms for timbre- and rhythm-based multi-resolution audio classification. In: 1st Annual Music Information Retrieval Exchange. (2005)
18. Schapire, R.E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. *Machine Learning* **39**(2/3) (2000) 135–168
19. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**(2) (1989)
20. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA (1984)
21. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/MLRepository.html> (1998)
22. Diez, J.J.R., González, C.A.: Applying boosting to similarity literals for time series classification. In Kittler, J., Roli, F., eds.: *Multiple Classifier Systems*. Volume 1857 of *Lecture Notes in Computer Science*, Springer (2000) 210–219
23. Boné, R., Assaad, M., Crucianu, M.: Boosting recurrent neural networks for time series prediction. In Pearson, D., Steele, N.C., Albrecht, R.F., eds.: *Artificial Neural Networks and Genetic Algorithms*. Springer (2003) 18–22