Nonlin. Processes Geophys., 18, 81–89, 2011
www.nonlin-processes-geophys.net/18/81/2011/
doi:10.5194/npg-18-81-2011

Nonlinear Processes in Geophysics

Hidden semi-Markov Model based earthquake classification system using Weighted Finite-State Transducers

M. Beyreuther and J. Wassermann

Dept. of Earth and Environmental Sciences (Geophys. Observatory), Ludwig-Maximilians-Universität München, Germany

Received: 22 September 2010 – Revised: 17 January 2011 – Accepted: 18 January 2011 – Published: 14 February 2011

Abstract. Automatic earthquake detection and classification is required for efficient analysis of large seismic datasets. Such techniques are particularly important now because access to measures of ground motion is nearly unlimited and the target waveforms (earthquakes) are often hard to detect and classify. Here, we propose to use models from speech synthesis which extend the double stochastic models from speech recognition by integrating a more realistic duration of the target waveforms. The method, which has general applicability, is applied to earthquake detection and classification. First, we generate characteristic functions from the time-series. The Hidden semi-Markov Models are estimated from the characteristic functions and Weighted Finite-State Transducers are constructed for the classification. We test our scheme on one month of continuous seismic data, which corresponds to 370 151 classifications, showing that incorporating the time dependency explicitly in the models significantly improves the results compared to Hidden Markov Models.

1 Introduction

The automatic detection and classification of seismic signals is increasing in significance since data centers have moved from the acquisition and archiving of single data snippets to streaming continuous seismic waveforms. Automatic detection and classification of earthquakes is used, for example, to automatically acquire consistent earthquake catalogues at volcanoes, to achieve class-dependent pre-selection of localization methods, and to exclude quarry blasts from earthquake catalogues. Our choice to reach a robust detection and classification algorithm is to adopt Hidden Markov Models (HMMs). This technique is very successfully applied to speech recognition (Young et al., 2002), and has practical applications to the field of seismology (Ohrnberger, 2001; Beyreuther and Wassermann, 2008) and other fields (Kehagias and Fortin, 2006). The advantages and disadvantages of this technique compared to other approaches is thoroughly covered in Beyreuther and Wassermann (2008) for the field of seismology.

HMMs provide a powerful tool to describe highly variable time series based on double stochastic models. The model acts on characteristic functions or features (estimated from the seismogram), e.g. the envelope or the power in different frequency bands, which describe the earthquake better than the pure ground motion signal itself. One part of the stochastic model represents the time dependency of these derived characteristic functions; the other part represents the distribution of the characteristic functions itself. Since this is a fully probabilistic approach, a confidence measure is naturally also provided. However, a drawback when using HMMs is that the probability of the duration for a single part in the HMM (called state) is an exponentially decaying function in time which is an unrealistic representation for the duration of earthquake classes or speech (Oura et al., 2008). To overcome this limitation, we apply Hidden semi-Markov Models (HSMMs) which use the more realistic Gaussians as state duration probability distributions.

The commonly used HMM decoding/classification technique (Viterbi algorithm) cannot be applied to HSMMs, as it relies strongly on the intrinsic HMM design. Therefore, we construct Weighted Finite-State Transducers (WFSTs) from the HSMMs for the purpose of classification (Mohri et al., 2002). This step also allows a much more flexible model refinement in the actual classification.

Correspondence to: M. Beyreuther
(beyreuth@geophysik.uni-muenchen.de)

Published by Copernicus Publications on behalf of the European Geosciences Union and the American Geophysical Union.

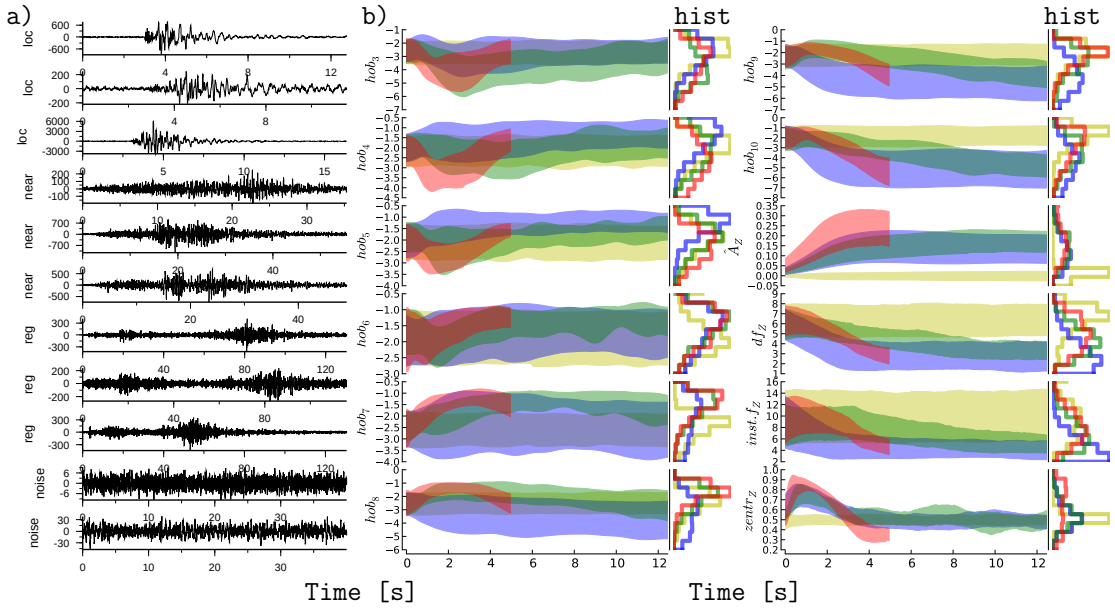

Fig. 1. Training data examples **(a)** and characteristic functions **(b)** generated from the training data set (plus/minus one standard deviation around the mean). Color-coding: reg in blue, near in green, loc in red, noise in yellow. A histogram over the timespan is appended vertically for each characteristic function in **(b)**. 3 s were subtracted from the start and added to the end of each sample in **(a)** to emphasize the contrast to the noise.

In the next section the HMMs are briefly introduced to the field of seismology, followed by a more detailed description of their HSMM extension and the corresponding WFST classifier. In order to show the potential for earthquake detection and classification, we apply HSMM as well as HMM to a one month continuous seismic dataset.

2 Theory

2.1 Hidden Markov Models

Hidden Markov Models (HMMs) are estimated from a training data set, i.e. they belong to the class of supervised classification techniques. The models are not operating on the seismic signal itself but on characteristic functions (also called features) generated from the seismograms which better represent the different classes of earthquakes.

Figure 1a shows an example of a training data set in the time-amplitude space. The label corresponds to the class name where the classes simply differ in epicentral distance (reg 100 km – 600 km, near 10 km – 100 km, loc 0 km – 10 km, noise noise). Note that the near and reg class have a quite similar signature in the time-amplitude space and therefore are not easy to distinguish. A much better characterization of the classes is shown in Fig. 1b, where different characteristic functions are plotted. Each band corresponds to a plus minus one standard deviation band around the mean of the characteristic function amplitude over all available training data for that specific class. In this representation it is much easier to distinguish the different classes, as can be seen by comparing the reg and near class (blue and green line) in the characteristic function space and in the amplitude space, respectively.

Figure 1b also makes clear why it is important to include time dependency in the model. If time dependency is excluded, the classes can only be distinguished through their histogram (as appended vertically for each characteristic function). As an example the half octave band 10 (hob_{10}) may be taken: The reg and near class (blue and green line) are not easily distinguished in the time independent histogram (Fig. 1b). However they are easily distinguishable in the time dependent characteristic function itself. For more details on the use of characteristic functions as well as the selection process, see Ohrnberger (2001) or Beyreuther and Wassermann (2008).

Figure 2 shows a sketch of a HMM for a single earthquake class. The observation is usually a vector containing multiple features, though for visualization purposes we use a single sample point ($\mathbf{o} = o$) which represents the absolute amplitude. The HMM segments the earthquake signal in different parts over time i (called states). For each part an observation probability distribution $b_i(\mathbf{o})$ of the characteristic function (here the absolute amplitude) is estimated. The sequence of the different parts is controlled by the (state) transition probability (a_{ij}), from part (i) to part (j). The probability of

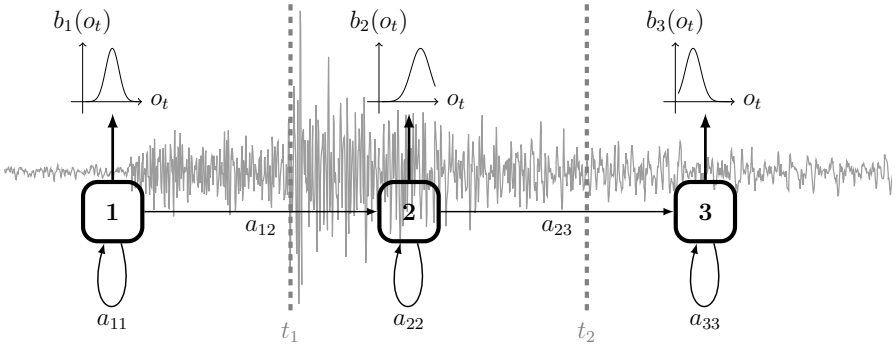

Fig. 2. Hidden Markov Model with three different states. The state transition probabilities are denoted with a_{ij}. The observation distribution for each state is denoted with $b_i(o_t)$.

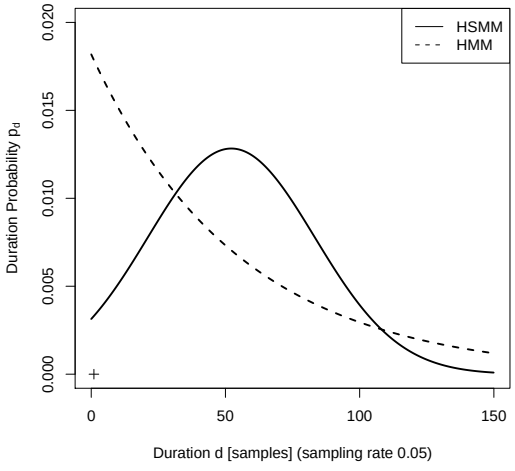

Fig. 3. Exponential (HMM) versus Gaussian (HSMM) state duration probability distribution.

the observation sequence $\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \mathbf{o}_4, \mathbf{o}_5 ...$ can then be calculated by taking the product of observation probabilities and state transition probability for each observation sample, e.g.

$$b_1(\mathbf{o}_1)a_{11}b_1(\mathbf{o}_2)a_{12}b_2(\mathbf{o}_3)a_{23}b_3(\mathbf{o}_4)a_{32}b_2(\mathbf{o}_5)..., \quad (1)$$

assuming that the state transition sequence is known/given. However, as the transition sequence is usually not known, the transition sequence which has the maximum probability is taken. Searching all possible transition sequences is incredibly time consuming and therefore usually an optimized algorithm (called Viterbi algorithm, see Rabiner, 1989) is used for classification.

In the HMM training procedure, the transition probabilities (a_{ij}) and the observation distributions (b_i) are estimated automatically through an expectation-maximization algorithm. For more details on the theory of HMMs, see Rabiner (1989) and Young et al. (2002).

2.2 The Extension to Hidden semi-Markov Models

The duration probability distribution of a particular HMM state is not included in the intrinsic HMM design. Nonetheless, according to the definition of HMMs the probability of staying T time steps in state i can be calculated from the state transition probabilities (a_{ij}) as follows:

$$a_{ii}^T(1-a_{ii}) = (1-a_{ii})\exp(T\log(a_{ii})) = \text{const}\cdot\exp(-T|\log(a_{ii})|) \quad (2)$$

with a_{ii} being the self-state transition probability ($0 \leq a_{ii} \leq 1$), which yields the negative logarithm in the third part of the equation. The result is an exponentially decaying function of the duration time T. An exponentially decaying function, however, is not an adequate representation of the duration distribution of certain states (e.g., P-wave and P-coda, S-wave and S-coda etc.) of an earthquake class, as this would imply that this part (state) has most likely length one or zero (see Fig. 3).

An alternative representation is to integrate the state duration probability distributions explicitly into the HMM. This HMM extension is known as a Hidden semi-Markov Model (HSMM) (Oura et al., 2006, 2008). In doing so, we are now able to approximate the duration probabilities through Gaussians. The HSMMs were a breakthrough in speech synthesis because it is crucial that certain speech parts have the correct duration since they otherwise sound unnatural (Zen et al., 2004).

Figure 4 provides an example of a three state Hidden semi-Markov Model. First a sample d_1 is drawn from the duration probability distribution (Gaussian) of the first state δ_1. Depending on the value of d_1, d_1 observations $\mathbf{o}_1 ... \mathbf{o}_{d_1}$ are generated with the corresponding probabilities $b_1(\mathbf{o}_1)...b_1(\mathbf{o}_{d_1})$. Then the second state is entered and the same procedure is continued for all remaining states. The corresponding probabilities are multiplied:

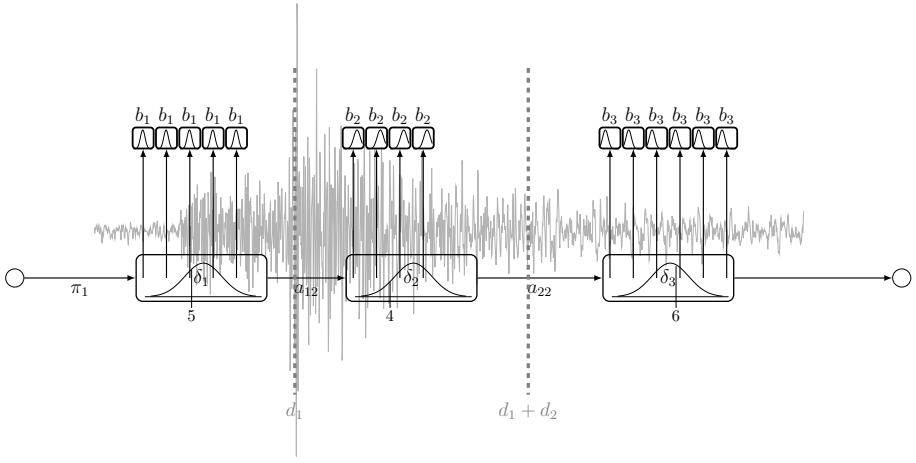

Fig. 4. Hidden semi-Markov Model. The duration probabilities $\delta_{1,2,3}$ are plotted in the states (lower rectangles). Directly when a state is entered, a sample is drawn from δ, e.g. five in the part. Consequently five observations are generated from the observation probability b_1 and then the next state is entered.

$$\delta_1(d_1)\prod_{i_1=1}^{d_1} b_1(\mathbf{o}_{i_1}) \quad \delta_2(d_2)\prod_{i_2=d_1}^{d_1+d_2} b_2(\mathbf{o}_{i_2})$$

$$\delta_3(d_3)\prod_{i_3=d_1+d_2}^{d_1+d_2+d_3} b_3(\mathbf{o}_{i_3}); \quad \text{with} \quad d_1+d_2+d_3=T. \tag{3}$$

The probability of the single HSMM given an observation sequence $\mathbf{o}_1 ... \mathbf{o}_T$ is the one with the values of d_1, d_2, d_3 which maximize Eq. (3). Again the observation distributions (b_i) and the duration probability distribution (δ_i) are estimated automatically during HSMM training (for details see Oura et al., 2006).

2.3 Weighted Finite-State Transducer decoding

The Viterbi algorithm is standard in HMMs classification (also called decoding). However, this cannot be applied for HSMMs because its optimized dynamic programming core relies strictly on the architecture of the HMM state transitions. Viterbi decoding is optimized and therefore extremely fast. In contrast, Weighted Finite-State Transducers (WFSTs) unfold the complete HSMM structure, where each unfolded path is modeled individually at the expense of speed. However, by associating weights such as probabilities, durations, or penalties to each possible path, the WFSTs provide a unified framework for representing and refining various models (used by Mohri et al., 2002, for speech and language processing). WFSTs are the standard HSMM decoder used in speech recognition (see Oura et al., 2006) and in following we show how to build up an HSMM decoder with a WFST.
Figure 5 shows the first five possible paths for the first state of the loc model. In total, the number of paths per state in our application ranges between 24-875; however, for visualization purposes, we show the first five only. The five observations in Fig. 4 (first "loc" state) are represented by the lowest path in Fig. 5. The other paths (Fig. 4), with 4,3,2 and 1 arcs, would correspond to the drawn values 4,3,2 and 1 of δ_1. The circles in Fig. 5 correspond to numbered nodes which the transducer can pass. All the probabilities are stored in the arcs and described by the corresponding labels. The annotation "7:−0" , for instance corresponds to the WFST input label 7 (the first of loc, see Table 1), no output label ("–") is assigned, and the negative log probability for this transition is 0 which corresponds to $\exp(-0) = 100\%$. The annotation "7:loc[1]/8.032" assigns the output label "loc[1]" to the path (in principle any name can serve as a label; we chose the name of the HSMM state as output label) and a negative log

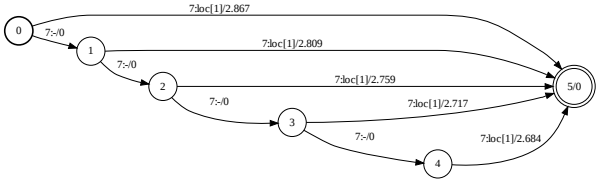

Fig. 5. The Weighted Finite-State Transducer for the first 5 time samples in the loc model. The lowest path (with 5 arcs) corresponds to the first part in Fig. 4 which is bound to the right by the dashed line labeled d_1. While the nodes only mark the steps in time, the arcs contain the input states, output labels, and the corresponding weight, which in this case is the negative log probability of the duration probability.

Table 1. The mapping of HSMM states and WFST input labels. Each earthquake class has three states.

HSMM state	reg 1	reg 2	reg 3	near 1	near 2	near 3	loc 1	loc 2	loc 3	noise 1
WFST input label	1	2	3	4	5	6	7	8	9	10

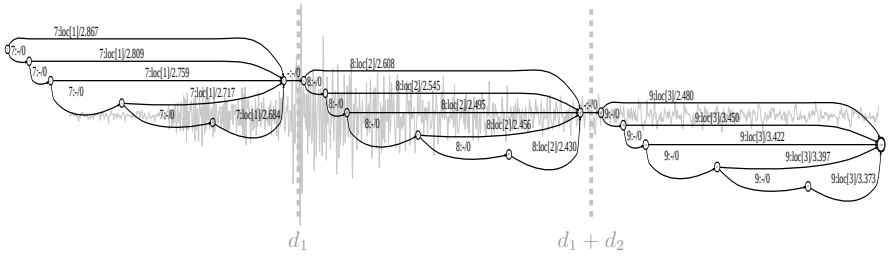

Fig. 6. Concatenated Weighted Finite-State Transducer. The WFST is the result of concatenating the WFST shown in Fig. 5 for the first, second and third loc state. The labeling of the above figure is described in Fig. 5. The corresponding mapping from HSMM states to WFST input labels is described in Table 1. Note that for visualization, only the first 5 paths per HSMM state are plotted.

probability of this transition of 8.032 which corresponds to $\exp(-8.032) = 0.033\%$. Therefore, all we need to do to build an earthquake classifier for the first loc state with a WFST is to link the input label to an HSMM state (Mohri et al., 2002) and to assign the negative log probability of the last arcs in the transducer to the duration probability of all previous arcs, thus setting their probability to 100%. The probability for staying two samples in state one (the second path from the top in Fig. 5) corresponds to $cdf(2) - cdf(1)$, whereby the cumulative duration distribution function is denoted as cdf.

The WFSTs allow composition, unification and concatenation of the transducers (for more details, see Mohri et al., 1997). In order to build up a WFST for the complete loc earthquake we simply concatenate the transducer for the first, second, and third loc state into one overall loc earthquake transducer. Figure 5 shows the transducer for the first loc state and Fig. 6 represents the concatenated transducer for the whole loc model.

The classifier for the reevaluation of the training data is constructed by unifying the concatenated WFSTs of the earthquake classes and the noise as shown in Fig. 7. In the classification process the negative log-likelihoods are minimized to find the most likely path. The WFST for the reevaluation of the training data in the next sections has in total 10^{16} different possibilities/paths.

The flexibility in constructing the WFST easily allows the introduction of minimum and maximum length criteria for the earthquake parts. By, for example, deleting the arcs pointing from nodes $(0 \rightarrow 5)$ and $(1 \rightarrow 5)$ in Fig. 5, the resulting WFST has a minimum duration of two samples. In the classification of an earthquake it should be impossible to directly travel from HSMM state $(1 \rightarrow 2 \rightarrow 3)$ and stay during all the remaining samples (order of 100 or 1000) in the third HSMM state, which then basically resembles only the third HSMM state. In order to avoid this behavior, and to increase the speed of the application, we build the WFST with the minimum length being the value of the 30th percentile and the maximum length being the 70th percentile of the duration distribution for each state, respectively.

3 Application

The HSMM detection and classification system was applied to a one month period (2007-09) of continuous data from the seismic station RJOB of the Bavarian Earthquake Service (http://www.erdbeben-in-bayern.de). The specified data set was chosen because the target classes loc, near and reg, which served as examples in the theory section and are represented and differ by epicentral distance. This enables us to easily decide whether a classification is correct or not, thus allowing a better evaluation of the advantages and disadvantages of the proposed algorithms. The training data set consists of 122 loc, 37 near, 115 reg and 176 noise events.

A first performance test is the reevaluation of the training data set. The models are trained from the training data and then as a second step, the same training data are then reclassified by using these trained models (allowing only one class per event as shown in Fig. 7). The results are shown in Fig. 8.

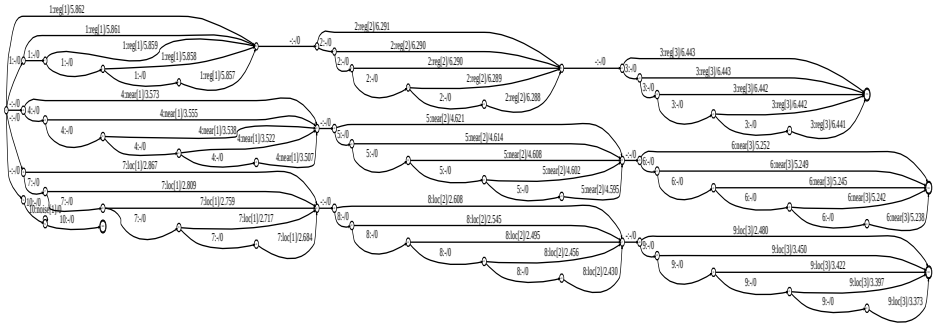

Fig. 7. Classifier design for the reevaluation of the training data. By concatenating the different HSMM states, one WFST per earthquake class is constructed (Fig. 6). The separate WFST for each class can then be unified (i.e. put in parallel) which results in one large WFST shown here. Note again that for visualization, only the first 5 paths per HSMM state are plotted.

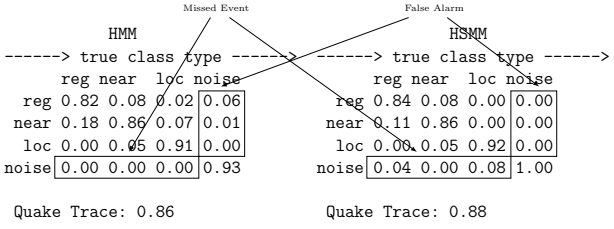

Fig. 8. Reevaluation of the training data. To the right the true class type is plotted; to the bottom the recognized one. E.g., for the HMM reg class, 82% of the data are correctly classified as reg and 18% are confused as near.

The detection performance of HMM versus HSMM (Fig. 8) is expressed by the rate of false alarms and missed events. The missed event rate is higher for HSMM, whereas the false alarm rate is higher for the HMM. A higher false alarm rate is problematic, because during continuous classification mostly noise is classified (e.g. 370 151 noise classifications in the following continuous period). Thus, even a low percentage of false alarms lead to a high total number. Nonetheless, the classification performance of both HMM and HSMM is similar. The reevaluation of the isolated training data set provides an indication of the performance, since over-training and performance in various noise conditions is not covered. Therefore we also chose to classify one month of continuous data.

In one month 370 151 classifications are made ($3600 \cdot 24 \cdot 30/7$, with 7 s window step size). The continuous data are classified in a sliding window of class dependent length (loc 14 s, reg 157 s, near: 47 s), with a window step of 7 s. The window length was chosen such that each class fits about twice inside the window. For each window and type, a [noise, earthquake, noise] and an only [noise] sequence are classified. This allows a flexible position of the earthquake in the window and calculation of the confidence measure P(earthquake)/P(noise), with P being the probability.

Figure 9 displays the results of the HMM and HSMM classification. The plot DATA shows three local earthquakes; the plots HSMM and HMM show the results of the two different models, respectively. The log probabilities of the earthquake models are plotted as colored bars, and the log probabilities of the corresponding noise model are plotted as dashed lines in the same color.

An earthquake is detected if the probability for the earthquake is greater than the probability for noise. The corresponding type is classified by selecting the earthquake class with the highest confidence measure, which in Fig. 9 corresponds to the vertical distance of the associated noise probability to the center of the earthquake probability (displayed as a bar). Other classifications in that period are deleted, even if they are longer than the class with the highest confidence measure. This procedure is known as "the winner takes all" (see e.g. Ohrnberger, 2001). By comparing the results from the HMM and HSMM classification, it is easy to see that the classes are much better characterized by the HSMM. Through their rather simple design the HMM even missclassifies the second and third local earthquake. It is also clear to see that the reg HSMMs have more realistic minimum duration (the 10 s duration of the reg HMM for the third local earthquake is impossible) due to the more flexible model design in the WFST.

This classification procedure is applied to a one month period of continuous data. In order to avoid a large number of false alarms, minimum confidence thresholds for a classification are introduced (for details see Beyreuther et al., 2008), which are chosen in such a way that the false alarm rate for both HMM and HSMM are similar. The results are shown in Table 2.

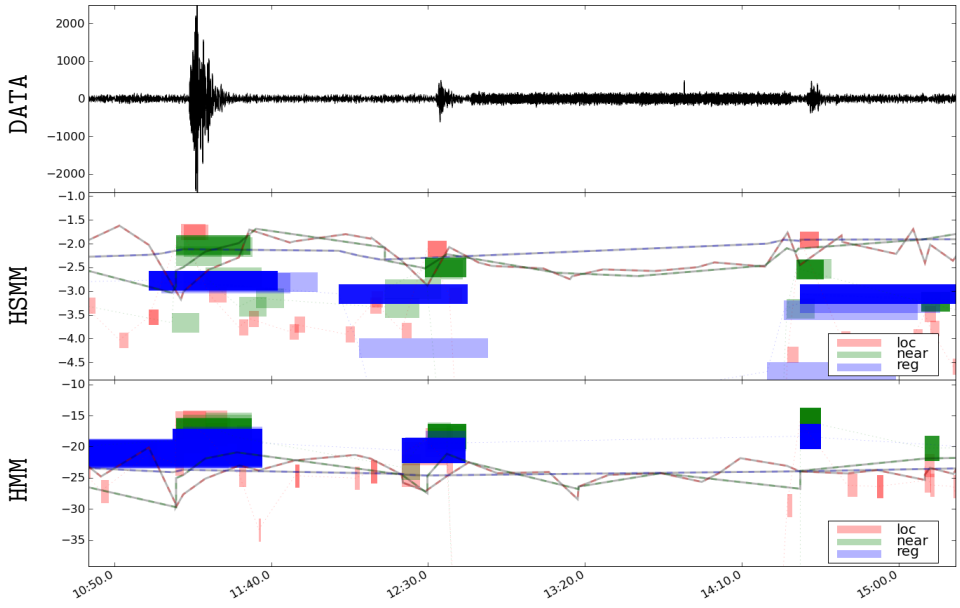

Fig. 9. The top plot shows three local earthquakes. The second and third plot shows the resulting log probabilities from HMM and HSMM detection and classification respectively. Colored bars denote the log probability of the earthquake classes, while the dashed lines denote the corresponding noise classes for each earthquake class.

Table 2. Results of one month of continuous classification. Confused events are events which are misclassified to another earthquake class. The confused events for the noise class correspond to the false alarms. For the comparison, the post-processing thresholds for both HSMM and HMM are chosen such that the false alarm rate stays for both algorithms at about two false alarms per day.

HMM	# of reg		# of near		# of loc		# of noise
correct events:	9	(26%)	4	(25%)	56	(67%)	370 084
confused events:	5	(14%)	1	(6%)	12	(14%)	67
missed events:	21	(60%)	11	(69%)	15	(18%)	–
total:	35		16		83		370 151
HSMM	# of reg		# of near		# of loc		# of noise
correct events:	23	(66%)	10	(63%)	75	(90%)	370 087
confused events:	5	(14%)	2	(13%)	5	(6%)	64
missed events:	7	(20%)	4	(25%)	3	(4%)	–
total:	35		16		83		370 151

The HSMM clearly outperforms the HMM. However, the results of the HMM and HSMM could be easily improved by using a different parameterization, e.g. more states. But since the computation time of the WFST decoding would then increase dramatically, we choose to use a rather simple setup. The reasons for choosing certain parameters and the parameters itself are discussed in the following.

4 Discussion

A key point for the HMM and HSMM design is the refinement to the correct parameter set. The parameters are the number of Gaussians which compose the observation probabilities (called Gaussian mixtures), the number of states in general, the model topology and the possibility of tying variances etc. The motivation for choosing the parameters and the parameters themselves are explained in the following.

1. We use left to right models as explained in the theory section due to the causality of the earthquakes (Fig. 1). For noise, which is clearly time independent (Fig. 1), only one state is used and thus no time dependency is assumed. Due to the limited amount of training data, it is statistically likely to over-train the model parameters. Consequently for a robust estimation of the parameters, only a limited amount of states and Gaussian mixtures for the observation probability $b_i(\mathbf{o})$ can be used. Also, the high computational costs of the WFST classification limits us to a maximum amount of six states (0.5 h CPU time for 1 h of data).

2. Because of the time-dependent characteristics of the earthquakes themselves (Fig. 1), we use more discretizations in time (6 states) for the earthquake models and only one for the noise model.

3. For the noise model we choose a higher number of Gaussian mixtures (4 mixtures) for the observation probability distributions $b_i(\mathbf{o})$ such that a large amount of different noise sources could be matched. For the earthquake models we use a single mixture.

4. In previous studies day noise and night noise classes were used, which match the anthropogenic noise during day time and the non-anthropogenic noise during night time (Ohrnberger, 2001; Beyreuther and Wassermann, 2008). In this study, however, we use only one noise class in order to avoid competing day noise and night noise models through the competitive training of Young et al. (2002); Zen et al. (2009).

5. The training of Young et al. (2002) and Zen et al. (2009) allows the computation of full multidimensional covariance matrices for the observation probability. However, in order to keep the number of free parameters low, the characteristic functions are first transformed to their principal axes (using principal component analysis), such that only the diagonal of the covariance matrices need to be used.

We also tried other parameter combinations: earthquake states i from three to six, Gaussian mixtures of $b_i(\mathbf{o})$ from one to six, and tied variances for all models. Clearly a lower number of earthquake states results in a lower number of discretizations in time and thus a worse classification result. More interestingly, a higher number of Gaussian mixtures for the observation probability distributions of the earthquake models did not achieve better results. This may indicate that the characteristic functions of the earthquakes are well represented by one Gaussian distribution. Also a higher amount of mixtures for the noise model did not achieve better results; four mixtures seem to be robust.

5 Conclusions

Seismology is a data rich science which allows nearly unlimited access to measures of ground motion. The migration of data centers' data acquisition from event based data snippets to continuous data has increased the amount of available waveforms dramatically. However, the target waveforms (earthquakes) are hard to detect, particularly with low signal to noise ratios. Advanced detection and classification algorithms are required, e.g. for automatically acquiring consistent earthquake catalogues for volcanoes, for class-dependent pre-selection of localization methods, or for excluding explosions from earthquake catalogues.

In this study we applied Hidden Markov Models, which are double stochastic models known from speech recognition for classification. We demonstrated the benefit of including a more realistic time dependence in the model (HSMM). The classification of one month of raw continuous seismic data (in total 370 151 classifications) shows a classification performance increase up to 40% using HSMM vs. HMM. However the performance increase has to be balanced with the increase of CPU time by a factor of 10 (1/2 h CPU time for 1 h data), thus making it difficult to process large data archives. For future work either a much more optimized WFST implementation needs to be used or the HMM models need to be refined in such a way that the natural duration of the earthquake parts is better represented.

This paper shows the improved performance from including more realistic time dependencies specifically for HMM. However, an increased performance should also be possible by including the time dependency in the model design of other supervised learning techniques such as support vector machines (Langer et al., 2009) or artificial neuronal networks (Ibs-von Seht, 2008) when classifying time series. Another major impact on the detection and classification performance, especially the false alarm rate, is easily achieved by combining the results of several stations, similar to the coincidence sums of common triggers.

In order to compare the results of this study to other methods in earthquake detection and classification (e.g. artificial neuronal networks, support vector machines, self organizing maps or tree based classifiers) benchmark data sets are required. Unfortunately these are currently not available for the field of seismology and, concluding the presented study, we think there is a strong demand for them.

Acknowledgements. The authors wish to thank Heiner Igel for his kind support. We also wish to thank the authors of the HMM Tool Kit (HTK), Young et al. (2002), the authors of the HMM-based Speech Synthesis System (HTS), Zen et al. (2009) and the authors of the AT&T FSM Library^{TM}, Mohri et al. (1997) for providing and developing such powerful software packages. The suggestions of three referees, Wu Ye and two anonymous reviewers, substantially improved the manuscript.

Edited by: R. Gloaguen
Reviewed by: W. Ye and two other anonymous referees

References

Beyreuther, M. and Wassermann, J.: Continuous earthquake detection and classification using Discrete Hidden Markov Models, Geophys. J. Int., 175, 1055–1066, 2008.

Beyreuther, M., Carniel, R., and Wassermann, J.: Continuous Hidden Markov Models: Application to automatic earthquake detection and classification at Las Canadas caldera, Tenerife, J. Volcanol. Geoth. Res., 176, 513–518, 2008.

Ibs-von Seht, M.: Detection and identification of seismic signals recorded at Krakatau volcano (Indonesia) using artificial neural networks, J. Volcanol. Geoth. Res., 176, 448–456, 2008.

Kehagias, Ath. and Fortin, V.: Time series segmentation with shifting means hidden markov models, Nonlin. Processes Geophys., 13, 339–352, doi:10.5194/npg-13-339-2006, 2006.

Langer, H., Falsaperla, S., Masotti, M., Campanini, R., Spampinato, S., and Messina, A.: Synopsis of supervised and unsupervised pattern classification techniques applied to volcanic tremor data at Mt Etna, Italy, Geophys. J. Int., 178, 1132–1144, 2009.

Mohri, M., Pereira, F. C. N., and Riley, M.: General-purpose Finite-State Machine Software Tools, http://www2.research.att.com/~fsmtools/fsm/index.html, 1997.

Mohri, M., Pereira, F., and Riley, M.: Weighted Finite-State Transducers in Speech Recognition, Comput. Speech. Lang., 16, 69–88, 2002.

Ohrnberger, M.: Continous automatic classification of seismic signals of volcanic origin at Mt. Merapi, Java, Indonesia, Ph.D. thesis, Institut für Geowissenschaften, Universität Postdam, 2001.

Oura, K., Zen, H., Nankaku, Y., Lee, A., and Tokuda, K.: Hidden semi-Markov model based speech recognition system using weighted finite-state transducer, in: Acoustics, Speech and Signal Processing, 1, 33–36, 2006.

Oura, K., Zen, H., Nankaku, Y., Lee, A., and Tokuda, K.: A Fully Consistent Hidden Semi-Markov Model-Based Speech Recognition System, IEICE T. Inf. Syst., E91-D, 2693–2700, 2008.

Rabiner, L. R.: A tutorial on Hidden Markov Models and selected applications in speech recognition, Proceedings of the IEEE, 77, 257–286, 1989.

Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P.: The HTK Book, Tech. rep., Cambridge University Engineering Department, HTK Version 3.2.1, 2002.

Zen, H., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T.: Hidden semi-Markov model based speech synthesis, in: Procceedings of ICSLP, 2, 1397–1400, 2004.

Zen, H., Oura, K., Nose, T., Yamagishi, J., Sako, S., Toda, T., Masuko, T., Black, A. W., and Tokuda, K.: Recent development of the HMM-based speech synthesis system (HTS), in: Procceedings of the Asia-Pacific Signal and Information Processing Association, 1–10, 2009.