# ENFORCING SPARSITY, SHIFT-INVARIANCE AND POSITIVITY IN A BAYESIAN MODEL OF POLYPHONIC PIANO MUSIC

*T. Blumensath, M. Davies*

Queen Mary, University of London
Department of Electronic Engineering
Mile End Road, London E1 4NS, UK

## ABSTRACT

In this paper we develop a Bayesian method to extract individual notes from a polyphonic piano recording. The distribution of the note activation is non-negative and we therefore introduce a modified Rayleigh distribution to model this note behaviour. Sparseness of the note activation is achieved by a mixture distribution that is a mixture of a delta function and the modified Rayleigh distribution. The used learning rule requires integration over the note activations, which is done using a Gibbs Sampling Monte Carlo method. We analyse the behaviour of the algorithm using a simplified test signal as well as a real piano recording.

## 1. INTRODUCTION

Sparse signal representations (e.g.[1]) often lead to signal descriptions in terms of meaningful features. On the other hand, signal models that restrict all or some of the coefficients to be positive have also been proposed for many applications with the promise of extracting salient signal features [2]. From a Bayesian point of view, these constraints are part of prior knowledge about the model parameters and should therefore be represented in the prior distribution of those parameters. We take this view here and develop a model that uses both the non-negativity constraint as well as the sparseness constraint. The aim of the model is to analyse musical mixtures, however, other application domains are possible.

In analysing music signals we are often interested in which notes are played at which time instances as well as in a description of the individual notes. For most music, many notes can sound at once such that analysis is complicated. In a previous paper [3] we introduced a shift-invariant Sparse Coding formulation to solve this problem. In this paper, we extend the work in [3] by introducing a positivity constraint on the note activations. This is done by introducing a modified Rayleigh distribution which was found to closely model note activations in the analysed piano signal. The estimation of the note prototypes requires integration over the note-activations. This integration is not possible analytically. We therefore develop a Gibbs Sampling Monte Carlo approximation to approximate this integral.

In section 2 we introduce the model which describes the signal analysed. Here we state a generative signal model which is a linear mixture of note prototypes. We also state the probability distributions of model parameters. In particular we introduce a modified Rayleigh distribution to model the amplitude of active notes. In section 3 we develop the computational strategies which allows us to learn the model parameters and to infer note activation. This is achieved by the introduction of a Gibbs Sampler that draws samples from the note distribution. In section 4 the developed method is used to analyse an artificial signal with known properties as well as a real piano recording.

## 2. MODEL FORMULATION

A standard method of dealing with time-series in signal processing is to partition the signal into blocks. We denote a realisation of such a block by $\mathbf{x} \in \mathbb{R}^M$. These observation blocks are assumed to be generated by a linear summation of a small number of note prototypes $\mathbf{a}_k$ of length $L$. We assume that a piano reproduces a scaled version of the same waveform whenever a note is played.[1] A complication when dealing with time-series is that features such as notes can be located at arbitrary locations relative to the chosen observation block. In order to account for this uncertainty we model the observation with all possible shifted versions of the note prototypes. The model does not exactly describe the signal, so we assume an i.i.d. Gaussian error term, which also facilitates Bayesian analysis. We write this generative model as:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \epsilon, \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^N$ is the vector of coefficients that describes when and which notes are played and $\epsilon$ is a vector of i.i.d. observation noise. Both $\mathbf{A}$ and $\mathbf{s}$ are unknown. In this paper we concentrate on the problem of learning the matrix $\mathbf{A}$, i.e. the note prototypes, by marginalising over the unknown coefficients $\mathbf{s}$.

As note prototypes can occur at arbitrary positions in the observation block, the matrix $\mathbf{A}$ has to include prototypes $\mathbf{a}_k$ at all possible shifts. This is shown graphically for two notes of length three with $M = 4$ and $N = 12$ below.

$A =$

$$\begin{bmatrix} \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 & 0 & 0 & 0 \\ 0 & \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 & 0 & 0 \\ 0 & 0 & \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 & 0 \\ 0 & 0 & 0 & \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 \end{bmatrix} \tag{2}$$

The different notes are shown with stars and circles respectively and the subscripts label the samples of each note.

In music most notes do not occur most of the time so that $\mathbf{s}$ is zero with very high probability. We further assume that notes are independent a priori.

[1] This is clearly not true in practice, especially as notes can have different lengths. However, we have already shown that such a model can extract a great deal of information from the signal and finds note-like functions [3].

The physical mechanism in a piano always excites the piano strings in the same direction, such that the first excursion of the observed waveform of a piano note is also always in the same direction. This means that the coefficients $\mathbf{s}$ always have the same sign. As the note prototypes $\mathbf{a}_k$ and the coefficients $\mathbf{s}$ can be inverted together without changing the reconstruction we can, without loss of generality, assume $\mathbf{s}$ to be non-negative. Furthermore, in most music performances notes are played at similar amplitudes - otherwise louder notes would overshadow quieter ones, and these would then be inaudible. These considerations lead us to propose the distribution for non-zero coefficients $\mathbf{s}$ described in the following section.

### 2.1. The Modified Rayleigh Distribution

The Rayleigh distribution is given as:

$$p_R(s;\sigma_R) = \frac{1}{\sigma_R} s e^{-s^2/2\sigma_R} \tag{3}$$

for $s > 0$ and zero otherwise. This distribution can be easily extended to allow for a shift parameter $\mu$, and is then:

$$p_R(s;\mu,\sigma_R) = \frac{1}{\sigma_R}(s-\mu) e^{-(s-\mu)^2/2\sigma_R} \tag{4}$$

for $s > \mu$ and zero otherwise. This distribution is zero for all values smaller than $\mu$. In the problem introduced above this is not desired. We therefore introduce a modification of the above distribution, which we call here the modified Rayleigh distribution and define as:

$$p_{mR}(s;\mu,\sigma_R) = \frac{1}{Z_{mR}} s e^{-(s-\mu)^2/2\sigma_{mR}} \tag{5}$$

for $s > 0$ and zero otherwise. Note that this distribution is nonzero for all positive values of $s$. This distribution has the advantage that the marginalisation required in the Gibbs sampler (see below) is analytically tractable. An example of this distribution is shown in figure 1 (dashed line). The normalising constant for this distribution is:

$$Z_{mR} = \sigma_{mR} e^{-(\mu)^2/2\sigma_{mR}} + 0.5\mu\sqrt{2\pi\sigma_{mR}}(1+\mathrm{erf}(\frac{\mu}{\sqrt{2\sigma_{mR}^2}})) \tag{6}$$

where $\mathrm{erf}(\cdot)$ is the error function.

We compare this distribution to the histogram of note amplitudes in figure 1. Here we show the histogram of the note amplitude as recorded from the velocity value of a midi keyboard, i.e. an electronic keyboard which records the velocity with which keys are pressed during a musical performance. The histogram here shows the velocity values for the notes of a performance of Ludwig van Beethoven's Bagatelle No. 1 Opus 33. The dashed line in this figure is the graph of a modified Rayleigh distribution defined above, the dotted line is the standard Rayleigh distribution and the dash dotted line is the shifted Rayleigh distribution.

It is clear that the modified Rayleigh distribution introduced above fits the distribution of the note activations better than the other two distributions. For other data such as biomedical time-series, however, other positive distributions for the non-zero coefficients might be more appropriate. The modified Rayleigh distribution can in this case be readily replaced by a zero mean Gausian distribution restricted to positive values or a uniform distribution over some positive interval. Both of these distributions can be used in the Gibbs Sampler developed below. For these well known distributions the derivation of the required terms is relatively easy. We therefore concentrate on the presentation of the derivation of the algorithm for the more complicated modified Rayleigh distribution.

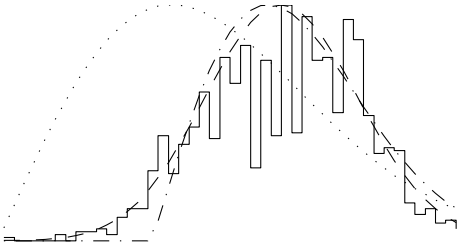

**Fig. 1**. Histogram of midi note velocities (solid) versus the modified Rayleigh distribution (dashed). Also shown are an unshifted Rayleigh distribution (dotted) and a shifted Rayleigh distribution (dash dotted).

### 2.2. Model Distributions

We use an i.i.d. Gaussian error term $\epsilon$ so that

$$p(\mathbf{x}|\mathbf{A},\mathbf{s}) \sim \mathcal{N}(\mathbf{x};\mathbf{A}\mathbf{s},\sigma_\epsilon \mathbf{I}). \tag{7}$$

We further define a factorial mixture prior $p(\mathbf{s}|\mathbf{u}) = \prod p(s_n|u_n)$ with

$$p(s_n|u_n) = u_n p_{mR}(s;\mu\sigma_p) + (1-u_n)\delta_0(s_n). \tag{8}$$

This is a mixture of a modified Rayleigh distribution and delta function at zero. The modified Rayleigh distribution models the amplitude of active notes while the delta function forces the coefficients to be exactly zero. The factorial form of the prior enforces the independence of the $s_n$. The hyper prior is the discrete distribution

$$p(u_n) = Z^{-1} e^{-0.5\lambda_u u_n}, u_n \in \{0,1\}, \tag{9}$$

where $Z$ is the appropriate normalising constant. We use the notation $\theta$ to denote the set of parameters $\theta = \{\mathbf{A}, \lambda_p, \lambda_\epsilon, \lambda_u, \mu\}$.

## 3. COMPUTATIONAL STRATEGY

### 3.1. Maximum Likelihood Estimation

Learning of the model parameters $\theta$ can be achieved by finding the maximum likelihood estimate of the marginal likelihood $\mathcal{Z} = p(\mathbf{x}|\theta) = \int p(\mathbf{x}|\theta,\mathbf{s})p(\mathbf{s}|\mathbf{u})p(\mathbf{u})\, d\mathbf{s}\, d\mathbf{u}$, which can be accomplished by stochastic gradient optimisation.

We can write the gradient of the marginal log likelihood with respect to the $p^{th}$ component of the coefficients $\theta$ and for a single observation as:

$$\frac{\partial \log \mathcal{Z}}{\partial \theta_p} = \left\langle \frac{\partial}{\partial \theta_p} \log p(\mathbf{x},\mathbf{s},\mathbf{u}|\theta) \right\rangle_{p(\mathbf{s},\mathbf{u}|\theta,\mathbf{x})}, \tag{10}$$

where $\langle\cdot\rangle$ denotes expectations.

### 3.2. Updating the Parameters

The gradient of $\log p(\mathbf{x}, \mathbf{s}, \mathbf{u}|\theta)$ with respect to the $p^{th}$ component of the feature $\mathbf{a}_k$ is:

$$\frac{\partial}{\partial a_{kp}} \log p(\mathbf{x}, \mathbf{s}, \mathbf{u}|\theta) = \frac{1}{\sigma_\epsilon} \sum_m \epsilon_m s_{k,p-m}. \tag{11}$$

Here we use $\epsilon_m = x_m - \sum_{k\in\mathcal{K}, l\in\mathcal{L}} a_{km+l} s_{kl}$. Note that this term is a convolution. The values of $\mathbf{A}$ can be updated by inserting this gradient into Eq. (10).[2] With this gradient the learning rules become:

$$\Delta a_{kp} = \frac{1}{\sigma_\epsilon} \left\langle \sum_m \epsilon_m s_{k,p-m} \right\rangle. \tag{12}$$

Here and in the following, expectations are always with respect to $p(\mathbf{s}, \mathbf{u}|\theta, \mathbf{x})$.

We find the updates for the other parameters in a similar way and get:

$$\Delta\lambda = -0.5 \left\langle \sum_{s_n \neq 0} (s_n - \mu)^2 - \frac{U}{c_1}(-0.5\mu\lambda^{-1}c_2 - c_3\lambda^{-2}) \right\rangle, \tag{13}$$

where the sum is over the non-zero $s_n$, U is the number of the non-zero $s_n$, $c_1 = \mu c_2 + \lambda^{-1}c_3$, $c_2 = 0.5\sqrt{2\pi\lambda^{-1}}(1+\mathrm{erf}(\mu\sqrt{0.5\lambda}))$ and $c_3 = e^{-0.5\lambda\mu^2}$,

$$\Delta\lambda_\epsilon = \left\langle \lambda_\epsilon^{-1} - \frac{(\mathbf{x} - \mathbf{A}\mathbf{s})^2}{\mu} \right\rangle, \tag{14}$$

$$\Delta\lambda_u = \left\langle \frac{1}{1 + e^{0.5\lambda_u}} - \frac{U}{N} \right\rangle \tag{15}$$

and

$$\Delta\mu = \left\langle \sum \lambda(s_n - \mu) - \frac{U}{c_1}c_3 \right\rangle, \tag{16}$$

where again, the summation is only over the non-zero $s_n$.

We can not evaluate these expectations analytically. Instead we propose Monte Carlo approximations with samples drawn using the Gibbs Sampling strategy described in the next section.

### 3.3. The Gibbs Sampler with the Modified Rayleigh/Delta Mixture Prior

The Gibbs Sampler proposed here is a modification of the method in [4]. The algorithm produces a Markov Chain by cyclic draws of samples from the conditional distributions $p(u_n|s_{j\neq n}, u_{j\neq n}, \mathbf{x}, \theta)$ and $p(s_n|u_n, s_{j\neq n}, \mathbf{x}, \theta)$. Here $s_{j\neq n}$ refers to all values of $\mathbf{s}$ apart from the $n^{th}$ so that we have to marginalise over $s_n$. This marginalisation can be solved analytically. We then set variable $u_n = 1$ with probability:

$$\begin{aligned} P(u_n = 1|s_{j\neq n}, \mathbf{x}, \theta) &= \frac{p(u_n=1|s_{j\neq n}, \mathbf{x}, \theta)}{\sum_{k=0}^{1} p(u_n=k|s_{j\neq n}, \mathbf{x}, \theta)} \\ &= \frac{1}{1+e^{-E_1}}, \end{aligned}$$

where

$$E_1 = -\log \frac{p(u_n = 1|s_{j\neq n}, \mathbf{x}, \theta)}{p(u_n = 0|s_{j\neq n}, \mathbf{x}, \theta)}. \tag{17}$$

[2] Note that the model used here has a scale ambiguity between the coefficients $\mathbf{s}$ and the functions $\mathbf{a}_k$. We therefore re-scale the functions $\mathbf{a}_k$ to unit $L_2$ norm after each update.

Therefore we only have to evaluate the logarithm of the ratio of the distributions such that the conditional distributions have to be known only up to a normalising term.

If we use the mixture prior in Eq. (8), the expression for $E_1$ in Eq. (17) becomes:

$$E_1 = -\frac{\lambda_{u_n}}{2} + \frac{\lambda_{n_n}}{2} b_n^2 + \ln\Phi \tag{18}$$

where $\Phi$ is:

$$\frac{Z_E}{Z_p}\left[\frac{1}{\Psi_n} e^{-0.5\nu^2\Psi_n} + 0.5\nu\sqrt{\frac{2\pi}{\Psi_n}}\left(1 + \mathrm{erf}\left(\nu\sqrt{\frac{\Psi}{2}}\right)\right)\right]$$

with

$$Z_E = e^{-0.5(-\nu^2\Psi_n + b_n^2\lambda_{n_n} + \mu_n^2\lambda)}$$

and

$$Z_p = \lambda^{-1} e^{-\mu_n^2 0.5\lambda} + 0.5\mu_n\sqrt{2\pi\lambda^{-1}}\left(1 + \mathrm{erf}\left(\mu_n\sqrt{0.5\lambda}\right)\right)$$

$\nu_n$ and $\frac{1}{\Psi_n}$ are the parameters of the posterior $p(s_n|s_{j\neq n}, u_n = 1, \theta)$ which is also of the modified Rayleigh form. They are given as:

$$\nu_n = \frac{\lambda_{n_n} b_n + \lambda\mu_n}{\lambda_{n_n} + \lambda}$$

and

$$\Psi_n = \lambda_{n_n} + \lambda$$

Here we have used the notation $\lambda_{n_n} = \|\mathbf{A}_n\|^2\lambda_\epsilon$ and $b_n = \frac{\mathbf{A}_n^T(\mathbf{I} - \mathbf{A}_{\S n=0})}{\|\mathbf{A}_n\|^2}$ where $\mathbf{A}_n$ is the n$^{th}$ column of the matrix $\mathbf{A}$ and $\lambda_\epsilon$ is the inverse of the variance of the likelihood.

Once we have sampled from $p(u_n|s_{j\neq n}, \mathbf{x}, \theta)$ we need to draw samples from $p(s_n|s_{j\neq n}, u_n, \mathbf{x}, \theta)$:

$$s_n \sim \begin{cases} p_{mR}(s; \nu_n, \Psi_n^{-1}) & \text{if } u_n = 1 \\ \delta_0(s) & \text{if } u_n = 0. \end{cases}$$

### 3.4. Sampling from the Modified Rayleigh Distribution

In the above sampling scheme it is necessary to draw samples from the modified Rayleigh distribution. this requires implementation of a method that allows us to draw samples from this distribution for different parameters. It is instructive to write the modified Rayleigh distribution in the form:

$$\frac{1}{Z_{mR}}(s) e^{-(s-\mu)^2/2\sigma_{mR}^2} = \frac{1}{Z_{mR}}\left((s-\mu)e^{-(s-\mu)^2/2\sigma_{mR}} + \mu e^{-(s-\mu)^2/2\sigma_{mR}}\right).$$

For $s > \mu$, the modified Rayleigh distribution can be understood as a mixture distribution of a Gaussian and a shifted Rayleigh distribution. However, the modified Rayleigh distribution is defined for values greater than zero, while the shifted Rayleigh distribution is defined for values greater than $\mu$, as it would be negative for values smaller than $\mu$. We therefore propose a hybrid sampling strategy, which first determines whether the value is greater or smaller than $\mu$. If $s > \mu$ we have

$$p(s > \mu) = p_1 = \frac{1}{Z_{mR}}(\sigma_{mR} + 0.5\mu\sqrt{2\pi\sigma_{mR}})$$

whilst for $s < \mu$ $p(s < \mu) = p_2 = 1 - p_1$.

With probability $p_1$, $s > \mu$ which means that we can sample from:

$$p(s|s > \mu) = \mu + \left[\frac{\sigma_{mR}}{Z_{mR}}\right] \sigma_{mR}^{-1}(s) e^{-0.5\sigma_{mR}^{-1} s^2}$$

$$+ \left[0.5\frac{\mu}{Z_{mR}}\sqrt{2\pi\sigma_{mR}}\right] 2\sqrt{\frac{\sigma_{mR}^{-1}}{2\pi}} e^{-0.5\sigma_{mR}^{-1} s^2},$$

which is a mixture of a rectified Gaussian and a shifted Rayleigh distribution with mixing probabilities given in the square brackets. For $s < \mu$ we have to reject samples smaller than 0. For $s < \mu$ we know that the distribution is bounded from above by

$$\frac{1}{Z_{mR}} \mu e^{-(s-\mu)^2/2\sigma_{mR}}$$

as

$$\frac{1}{Z_{mR}} (s-\mu) e^{-(s-\mu)^2/2\sigma_{mR}}$$

is negative. We therefore use a simple rejection sampler to draw samples for $0 < s < \mu$.

## 4. EXPERIMENTAL ANALYSIS

Our application of interest is to extract individual piano notes from a polyphonic recording of a piano performance. However, the computational complexity of this task is very high. Therefore, we study the proposed method on a simplified signal first, which allows better analysis of the method proposed. After that we tackle the problem of interest, however, we have to introduce a simplification step into the algorithm to be able to compute results for a real world signal.

### 4.1. A Toy Problem

We generated a test signal, which on the one hand has many of the properties of the signal of interest, but on the other hand requires calculation of a smaller number of parameters. This is done by using the information of a real piano performance of Ludwig van Beethoven's Bagatelle No. 1 Opus 33. This information includes the strength with which each note was played and the length, pitch and timing of each note. To generate a simple signal we restricted all pitches played to one octave (12 notes), and also reduced the time scale. However, the relative timing as well as the strength of the notes were left unchanged. The signal was then generated from features of length 128 samples. Thus the signal followed the model studied here in that it was generated from 12 waveforms at different locations and with different amplitudes. The location and the amplitudes of the waveforms had the same statistics as the original piano performance. We did not add noise to the model.

We used this signal and trained the model with 12 functions of length 128 samples. We initialised the functions to sinusoids of different frequencies. After 10000 iterations the functions had converged to those shown in figure 2. Figure 2 shows the original functions from lowest to highest frequency with dotted lines. These are labelled with letters from A to L. The learned functions are overlaid with solid lines. To assign the learned functions to the true functions we calculate the inner product between each of the learned functions and each of the true functions. We then assign to each true function the learned function with the highest inner product. These inner products are shown in table 1. Two of the learned functions did not have a high inner product with any of the

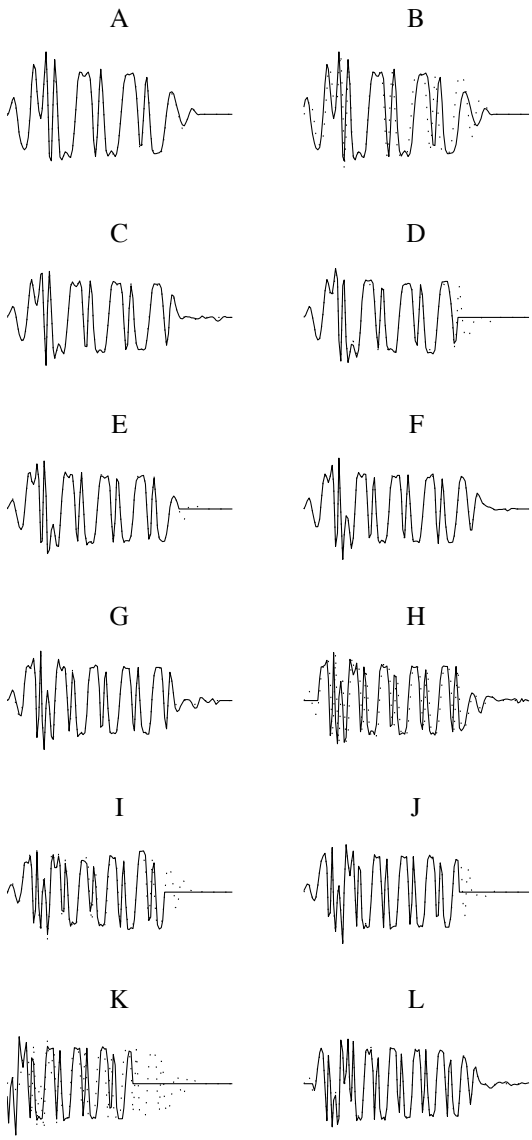


**Fig. 2**. Learned (solid) vs. original (dotted) functions.

true functions. Therefore pairs of true functions A and B, as well as true functions K and L, were assigned a single learned function each. Due to the use of a shift-invariant model we found that many of the learned functions were slightly shifted to the original functions. We also corrected for this in the assignment of learned to true functions. It is clear that the learned functions model the original functions relatively well.

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 0.9980 | 0.7106 | 0.9986 | 0.9721 | 0.9958 | 0.9987 |
| G | H | I | J | K | L |
| 0.9957 | 0.9942 | 0.9102 | 0.9719 | 0.4836 | 0.9966 |

**Table 1**. $L_2$ distance between the learned and original functions.

We see in table 1 that functions B and K are not represented

well by one of the learned functions. All other functions have been found with high accuracy. This lack of learning can be explained by the fact that functions B and K only occur a few times in the signal. However, whether the model has a local maximum or whether the learning for the missing features is very slow could not be determined.

The model parameters converged to $\lambda_\epsilon = 172$, $\lambda_p = 1.05$, $\lambda_u = 13.15$ and $\mu = 0.45$.

### 4.2. Learning Piano Notes: A Real World Problem

In this section we analyse a real piano recording. For a training signal we use the acoustic recording of the same piano performance which we used in the previous section. This time we learned 100 functions of length 1024 samples. The functions were initialised to i.i.d. Gaussian signals. As the dimension of the problem is too large for a straightforward implementation, we used a subset selection step to preselect a small number of functions before sampling. This selection process was dependent upon the correlation of the signal with the functions as described in [3]. The results in figure 3 show 47 functions which converged to harmonic signals after 100,000 iterations. The left panel shows the time domain structure of the functions while the right side gives the magnitude spectrum. The harmonic note like structure of the functions is clearly visible. The parameters were: $\lambda_\epsilon = 58$, $\lambda_p = 1.1$, $\lambda_u = 8.3$ and $\mu = -0.039$.

## 5. CONCLUSION

In this paper we have introduced a modified Rayleigh distribution to model the distribution of note amplitudes in a piano signal. This distribution was used in a mixture prior so that we enforced sparsity as well as positivity of note activations. A Gibbs Sampler was developed to draw samples from the note activation distribution conditioned on an observation. This sampler allowed us to learn note-like features from piano recordings as was demonstrated both with a toy problem as well as with a recording of a real piano performance.

Both, sparsity and non-negativity have been proposed to extract meaningful features from signals. We have shown that these constraints are easily implemented using Bayesian techniques. Other problems might require another prior formulation as used here, but it is easy to replace the proposed modified Rayleigh distribution with zero mean Gaussian or uniform distributions, which are restricted to positive values. The integrals required for the Gibbs Sampling method proposed here can then still be solved analytically.

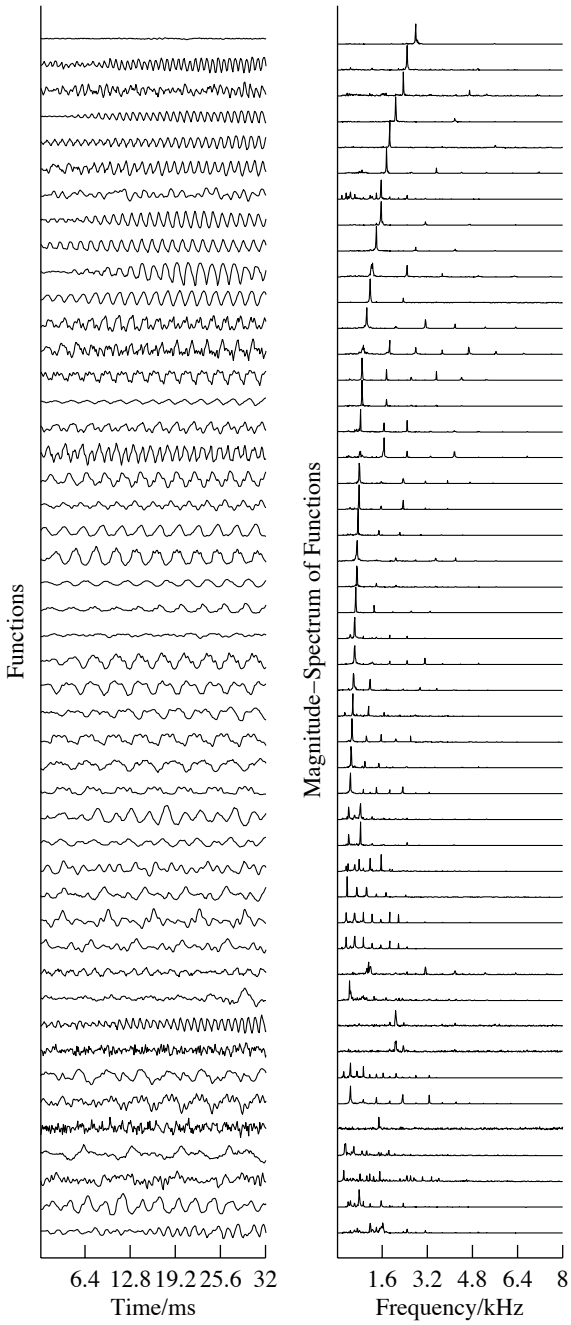


**Fig. 3**. Time domain representation (left) and magnitude spectrum (right) of the 47 harmonic notes learned.

## 6. REFERENCES


[1] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[2] D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 21, pp. 788–791, Oct 1999.

[3] T. Blumensath and M. Davies, "Unsupervised learning of sparse and shift-invariant decompositions of polyphonic music," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[4] P. Sallee and B. A. Olshausen, "Learning sparse multiscale image representations," in *Advances in Neural Information Processing Systems*. MIT Press, 2003, pp. 1327–1334.