Missing Data Imputation for Time-Frequency Representations of Audio Signals

Paris Smaragdis · Bhiksha Raj · Madhusudana Shashanka

Abstract With the recent attention towards audio processing in the time-frequency domain we increasingly encounter the problem of missing data within that representation. In this paper we present an approach that allows us to recover missing values in the time-frequency domain of audio signals. The presented approach is able to deal with real-world polyphonic signals by operating seamlessly even in the presence of complex acoustic mixtures. We demonstrate that this approach outperforms generic missing data approaches, and we present a variety of situations that highlight its utility.

1 Introduction

In this paper we address the problem of estimating missing regions of time-frequency representations of audio signals. The problem of missing data in the time-frequency domain occurs in several scenarios. For example, this problem is common in speech processing algorithms that employ computational auditory scene analysis or related methods to mask out time-frequency components for recognition, denoising or signal separation [1][2]. An increasing number of audio processing tools allows interactive spectral editing of audio signals, which can often result in the excision of time-frequency regions of sound. Aggressive audio compression techniques often introduce regions of "spectral holes". In yet other scenarios, such as in signals that have passed through a telephone or have encountered other linear or non-linear filtering operations, removal of entire time-frequency regions of the signal can occur naturally (for example through bandwidth reductions).

In a majority of these scenarios the goal is to resynthesize the audio from the incomplete time-frequency characterizations. To do so, the "missing" regions of the time-frequency representations must first be "filled in" somehow, in order to effect the transform from the time-frequency representation to a time-domain signal. In certain

Paris Smaragdis
Adobe Systems

Bhiksha Raj
Carnegie Mellon University

Madhusudana Shashanka
United Technologies Research Center

cases the missing values can be set to zero and the resulting reconstructions do not suffer heavily from perceptible artifacts. In most cases however a moderate to severe distortion is easily noticeable. This distortion can render speech recordings unintelligible, or severely reduce the quality of a music recording thereby distracting the listener.

Although existing generic imputation algorithms [3] can be used to infer the values of the missing data they are often ill-suited for use with audio signals and result in audible distortions. Other algorithms such as those in [1,4] are suitable for imputation of missing time-frequency components in speech, or in the case of [5] musical audio. However, these algorithms typically exploit continuity in spectral structures and are implicitly aided by the fact that the targeted recordings usually contain signals from either a single source (the voice), or repetitions of constant-state fixed spectra. General audio or music recordings however include a variety of sounds, many of which are concurrently active at any time, and each of which has its own typical patterns, and are hence much harder to model.

The algorithm proposed in this paper characterizes time-frequency representations as histograms of draws from a mixture model as proposed in [6]. Being a mixture, this model is implicitly capable of representing multiple concurrent spectral patterns an attribute that makes it a well suited for complex audio scenes, such as music or busy environments. The process of imputing missing values then becomes one of learning from incomplete data. Experimental evaluations show that the spectral constructions obtained with our algorithm result in distinctly better resynthesized signals than those obtained from other common missing data methods.

The remainder of this paper is organized as follows. In section 2 we introduce the exact problem we address with this approach, namely the problem of "holes" in time-frequency representations of audio signals and the shortcomings of current imputation techniques in dealing with them. In section 3 we describe the statistical modelling approach we employ and in section 4 we describe how our model can be estimated from and used to impute missing portions of incomplete spectrograms. Our model works on the magnitudes of time-frequency representations of audio. For the process of imputation of missing time-frequency terms to be complete, the phase too is required. In 5 we describe how we estimate the phase of imputed time-frequency components. Finally in sections 6 and 7 we describe our experiments and present our conclusions.

2 Missing data

In this section we describe the specific application domain that we are focusing on and present a few motivating examples. We then describe two of the standard tools used for missing data imputation, tools that we will be using as a baseline comparison to our proposed approach.

2.1 Missing data in the time-frequency domain

In this paper we will assume that the time-frequency representations are derived through short-time Fourier transformation (STFT) of the signal [7]. The short-time Fourier transform converts an audio signal into a sequence of vectors, each of which represents the Fourier spectrum of a short (typically 20-60ms wide) segment or *frame* of the signal. The STFT of a signal can be inverted to the original time-domain signal

by an inverse short-time Fourier transform. Being complex, the STFT of the signal has both a magnitude and a phase. However most sound processing algorithms for denoising, recognition, synthesis and editing operate primarily on the magnitude since it is known to represent most of the perceptual information in the signal – the phase contributes mainly to the perceived quality of the signal rather than its intelligibility. Since the phase reconstruction is highly dependent on the magnitude values we will primarily examine the reconstruction of the magnitudes of the missing time-frequency terms in this paper. Upon introducing our approach to impute the magnitude values we briefly show how we can easily find appropriate phase values.

Figure 1a shows an example of the magnitude of the STFT of a classical music recording that was contaminated by the ringing of a phone in the audience. We will refer to matrix-like representations of the magnitudes of STFTs of a signal, such as the one in Figure 1a as "magnitude spectrograms", or simply as "spectrograms" in this paper. Spectral magnitudes have the property that when multiple sounds co-occur, the spectral magnitudes of the mixed signal are approximately (but not exactly) equal to the sum of the spectral magnitudes of the component sounds. This is apparent in the spectrogram in Figure 1a which shows the distinctive spectral patterns of both the music and phone ring. Although in theory this additivity holds true for spectral *power* when the component signals are uncorrelated, in practice phase cancelations in finite analysis windows make this true for spectral magnitudes raised to a power that is closer to 1.0 than 2.0.

A spectrogram with "missing" data is one where some time-frequency components have been lost or erased due to some reason. Figures 1b, 1c and 1d show examples of spectrograms with missing data. In the first two examples time-frequency regions of the spectrogram have been erased to eliminate the phone ring from the signal, automatically in one case and by manual editing in the other. In the third example overzealous mp3 compression has removed much of the time-frequency content originally in the recording. Missing data may also occur for other reasons, such as systematic filtering, channel artifacts, etc. In order to reconstruct a time-domain signal from these incomplete spectrograms the values in the missing regions must be somehow filled-in. A simple technique is to simply floor these terms to some threshold value; however time-domain signals reconstructed from such spectrograms will often contain audible and often unacceptable artifacts. A more principled approach is required to reconstruct the missing regions in an acceptable manner.

2.2 Traditional Missing Data Approaches

The problem of replacing missing attributes of data has long vexed researchers, and a vast array of solutions have been proposed. The various solutions can briefly be summarized as those that impute missing attributes based on the assumed or estimated congruence of an incomplete data vector to other data for which the corresponding attributes are known e.g. [8], and those that replace the missing values based on local [9] or global statistical trends in the data [10][11].

Two successful techniques for imputing missing data attributes, that illustrate both the congruence-based and statistical approaches to imputation, are based on the nearest-neighbors and the Singular Value Decomposition (SVD) algorithms. In order to provide an introduction to this problem and since we will be using these approaches

as benchmarks against which we compare our approach we briefly describe them in this section.

In general, with missing data problems we assume that we have a matrix containing our data of interest which is missing some of its entries. There are multiple classifications of the missing data problem depending on the pattern of the missing entries, and whether that is conditional on their values . In our case we will assume that the missing entries are missing completely at random, i.e. these entries do not depend on the values of the observed or the missing data.

The nearest-neighbors algorithm is a very simple two-step process:

1. For each of the input's vectors that contain a missing value, compute the distance between them and all the other available vectors in that matrix. Do so using only the observed entries, and find the K nearest neighbors.
2. Impute the missing entries of each vector by averaging the corresponding elements from the K nearest neighbors that have these points available.

This can be seen a "local" technique which seeks similar looking areas to the one with the missing data, and uses their statistics to perform imputation. In doing so this approach ignores the global statistics of the input and because of that, it is able to gracefully handle inputs with complex structure and unusual samples. As we will see later on, when dealing with inputs which are known to be mixtures the nearest neighbor model does not have the mechanism to handle that structure and can often fail.

On the other end, the SVD model collects statistics from the entire input and finds solutions which are as predictable as possible. The steps in that approach are as follows:

1. Replace all missing values with an initial value. That can be either random values, or something more statistically relevant such as the mean of the input.
2. Compute the SVD of the resulting matrix and replace the missing values with their prediction according to the SVD decomposition.
3. Repeat this process until the change in the imputed missing data falls below some user-defined threshold.

In contrast to nearest-neighbors, this is a "global" approach. By performing an SVD we obtain information about the global statistics of the input and attempt to fill-in the missing data in such a way so that the input becomes statistically consistent. Because of this, missing data which are part of rare samples will be poorly approximated and be biased towards the form of the average input. On the other hand, for consistent inputs with a lot of missing values this approach can provide better averaging than the nearest neighbor model and provide less noisy estimates. It is also able to deal with mixed inputs since it employs a mixture model.

Both of these techniques are described in more detail in [12]. More advanced imputation algorithms have certainly been developed, but they are often specialized and not significantly better than the two above approaches. For this reason, and for illustrative purposes seen below, we will use these two algorithms as a benchmark when evaluating the performance of our proposed algorithm.

3 Proposed Approach

We will now present the model we will use for the problem at hand. We will start by explaining how we can think of spectrograms as scaled histograms and how that implies

a particular statistical model which is best suited for such data. We then define that model and present the estimation procedure one can use to find the model's parameters for a given input.

3.1 Modeling the Spectrogram

At the outset, we would like to specify the terminology and notation we will use. We will denote the (magnitude) spectrogram of any signal as $\mathbf{S}$. The spectrogram consists of a sequence of (magnitude) spectral vectors $S_t,\ 0 \leq t < T$, each of which in turn consists of a number of frequency components $S_t(f),\ 0 \leq f < F$. All of these terms, being magnitudes are non-negative.

In our model we view $\mathbf{S}$ as a scaled version of a histogram $\hat{\mathbf{S}}$ comprising purely integer valued components, such that $\mathbf{S} = \mathcal{C}^{-1}\hat{\mathbf{S}}$; however the scaling factor $\mathcal{C}$ cancels out of all equations in our formulation and is thus not required to be known. In the rest of the paper, we therefore treat $\mathbf{S}$ itself as a histogram and do not explicitly invoke the scaling factor, besides assuming that it is very large so that for any $\hat{S}_t(f)$ such that $S_t(f) = \mathcal{C}^{-1}\hat{S}_t(f)$ the following holds:

$$\mathcal{C}^{-1}\hat{S}_t(f) \approx \mathcal{C}^{-1}(\hat{S}_t(f) + 1). \tag{1}$$

3.2 Proposed Model Definition

We model each spectral vector S_t as a scaled histogram of draws from a mixture multinomial distribution. Per our model, S_t is generated by repeated draws from a distribution $P_t(f)$, where f is a random variable over the frequencies $\{1, \ldots, F\}$, and $P_t(f)$ is a mixture multinomial given by

$$P_t(f) = \sum_{z=1}^{Z} P_t(z)P(f|z) \tag{2}$$

Here z represents the identity of the Z multinomial components in the mixture. $P(f|z)$ are the multinomial components or "bases" that compose the mixture. Note that the component multinomials $P(f|z)$ are not specific to any given S_t but are the same at all t. $P(f|z)$ are thus assumed to be characteristic to the entire data set of which $\mathbf{S}$ is representative. The only parameter that is specific to t are the mixture weights $P_t(z)$.

The model essentially characterizes the spectral vectors themselves as additive combinations of histograms drawn from each of the multinomial bases. Consequently, it is inherently able to model complex sounds such as music that are additively composed by several component sounds. This is in contrast to conventional models used for data imputation *e.g.* [1][2][4], that model the spectral components as the outcome of a single draw from a distribution (although the distribution itself might be a mixture) and cannot model the additive nature of the data.

The model of Equation 2 is nearly identical to the one-sided model for Probabilistic Latent Semantic Analysis, introduced by Hoffman [13,14], with the distinction that whereas the original PLSA model characterizes random variables as documents and words, we refer instead to time and frequencies. Also, while the one-sided PLSA specifies a probability distribution over documents, in our model we do not have a similar probability distribution over the time variable t.

3.3 Learning the model parameters

We can estimate the parameters of the generative model in Equation 2 for $\mathbf{S}$ using the Expectation Maximization algorithm [14]. In the expectation step of the EM algorithm at each time t we estimate the *a posteriori* probabilities for the multinomial bases conditioned on the observation of a frequency f as:

$$P_t(z|f) = \frac{P_t(z)P(f|z)}{\sum_{z'=1}^{Z} P_t(z')P(f|z')} \tag{3}$$

In the maximization step we update the spectral-vector-specific mixture weights $P_t(z)$ and data-set characteristic multinomial bases $P(f|z)$ as:

$$P_t(z) = \frac{\sum_{f=1}^{F} P_t(z|f)S_t(f)}{\sum_{z'=1}^{Z}\sum_{f=1}^{F} P_t(z'|f)S_t(f)} \tag{4}$$

$$P(f|z) = \frac{\sum_{t=1}^{T} P_t(z|f)S_t(f)}{\sum_{f'=1}^{F}\sum_{t=1}^{T} P_t(z|f')S_t(f)} \tag{5}$$

4 Estimation with Incomplete Data

The algorithm of Section 3.3 assumes that the entire spectrogram $\mathbf{S}$ is available to learn model parameters. When the spectrograms are incomplete, several of the $S_t(f)$ terms in Equation 4 will be missing or otherwise unknown. Our objective in this paper is to estimate the missing components on the data. Along the way we will also estimate the model parameters themselves as necessary.

In the rest of the paper we use the following notation: we will denote the *observed* regions of any spectrogram $\mathbf{S}$ as $\mathbf{S}_o$ and the *missing* regions as $\mathbf{S}_m$. Within any spectral vector S_t of $\mathbf{S}$, we will represent the set of observed components as S_t^o and the missing components as S_t^m. $S_t^o(f)$ and $S_t^m(f)$ will refer to specific frequency components of S_t^o and S_t^m respectively. $\mathcal{F}_t^o$ will refer to the set of frequencies for which the values of S_t are known, *i.e.* the set of frequencies in S_t^o. $\mathcal{F}_t^m$ will similarly refer to the set of frequencies for which the values of S_t are missing, *i.e.* the set of frequencies in S_t^m.

4.1 The Conditional Distribution of Missing Terms

In the first step we obtain the conditional probability distribution of the missing terms $S_t^m(f)$ given the observed terms S_t^o and the probability distribution $P_t(f)$ from which S_t was drawn. Let $N_t^o = \sum_{f\in\mathcal{F}_t^o} S_t^o(f)$. N_t^o is the total value of all observed spectral frequencies at time t. Let $P_{o,t} = \sum_{f\in\mathcal{F}_t^o} P_t(f)$ be the total probability of all observed frequencies at t. The probability distribution of S_t^m, given that the frequencies in $\mathcal{F}_t^o$ are known to have been drawn exactly N_t^o times cumulatively is simply a *negative multinomial* distribution [15,16]:

$$P(S_t^m) = \frac{\Gamma(N_t^o + \sum_{f\in\mathcal{F}_t^m} S_t^m(f))}{\Gamma(N_t^o)\prod_{f\in\mathcal{F}_t^m}\Gamma(S_t^m(f)+1)} P_{o,t}^{N_t^o} \prod_{f\in\mathcal{F}_t^m} P_t(f)^{S_t^m(f)} \tag{6}$$

where $\mathcal{F}_t^m$ is the set of all frequency components in S_t^m. The expected value of any term $S_t^m(f)$ whose probability is specified by Equation 6 is given by[1]:

$$E[S_t^m(f)] = N_t^o \frac{P_t(f)}{P_{o,t}} \tag{7}$$

We now describe the actual learning procedures to estimate model parameters and missing spectral components. We identify two situations, stated in order of increasing complexity as (a) where the multinomial bases for the data $P(f|z)$ are known *a priori* and only the mixture weights $P_t(z)$ are unknown, and (b) where none of the model parameters are known. We address them in reverse order below to simplify the presentation.

4.2 Learning the Model Parameters from Incomplete Data

Let Λ be the set of all parameters $P_t(z)$ and $P(f|z)$ of the model defined by Equation 2. We derive a set of likelihood-maximizing rules to estimate Λ from $\mathbf{S}^o$ using the Expectation Maximization algorithm as follows.

We denote the set of draws that resulted in the the generation of $\mathbf{S}$ as $\mathbf{z}$. The *complete data* specification required by EM is thus given by $(\mathbf{S}^o, \mathbf{S}^m, \mathbf{z}) = (\mathbf{S}, \mathbf{z})$, where $\mathbf{S}^m$ and $\mathbf{z}$ are unseen. The EM algorithm iteratively estimates the values of Λ that maximizes the expected value of the log likelihood of the complete data with respect to the unseen variables [17], i.e. it optimizes:

$$\begin{aligned} Q(\Lambda, \hat{\Lambda}) &= E_{\mathbf{S}^m, \mathbf{z}|\mathbf{S}^o, \hat{\Lambda}} \log P(\mathbf{S}^o, \mathbf{S}^m, \mathbf{z}|\Lambda) \\ &= E_{\mathbf{S}^m|\mathbf{S}^o, \hat{\Lambda}} E_{\mathbf{z}|\mathbf{S}, \hat{\Lambda}} \log P(\mathbf{S}^o, \mathbf{S}^m, \mathbf{z}|\Lambda) \end{aligned} \tag{8}$$

where $\hat{\Lambda}$ is the current estimate of Λ. In this paper we won't take advantage of temporal continuity, thus we will treat the draws that compose any spectrum S_t independently of those that compose any other $S_{t'}$. Because of that Equation 8 simplifies to:.

$$Q(\Lambda, \hat{\Lambda}) = \sum_{t=1}^{T} E_{S_t^m|S_t^o, \hat{\Lambda}} E_{z_t|S_t, \hat{\Lambda}} \log P(S_t^o S_t^m, z_t|\Lambda) \tag{9}$$

[1] To be precise Equations 6 and 7 must actually be specified in terms of $\mathcal{C}^{-1} N_t^o + 1$; however, given the assumption in Equation 1, Equation 7, which is the primary equation of interest remains valid.

where z_t is the set of draws that composed S_t. Optimizing Equation 9 with respect to Λ, and invoking Equation 7 leads us to the following update rules:

$$P_t(z|f) = \frac{P_t(z)P(f|z)}{\sum_{z'=1}^{Z} P_t(z')P(f|z')} \tag{10}$$

$$N_t = \sum_{f \in \mathcal{F}_t^o} \frac{S_t^o(f)}{P_t(f)} \tag{11}$$

$$\begin{aligned} \bar{S}_t(f) &= S_t(f) \quad \text{if} \quad f \in \mathcal{F}_t^o \\ &\quad N_t P_t(f) \quad \text{if} \quad f \in \mathcal{F}_t^m \end{aligned} \tag{12}$$

$$P_t(z) = \frac{\sum_{f=1}^{F} P_t(z|f)\bar{S}_t(f)}{\sum_{z'=1}^{Z}\sum_{f=1}^{F} P_t(z'|f)\bar{S}_t(f)} \tag{13}$$

$$P(f|z) = \frac{\sum_{t=1}^{T} P_t(z|f)\bar{S}_t(f)}{\sum_{f'=1}^{F}\sum_{t=1}^{T} P_t(z|f')\bar{S}_t(f)} \tag{14}$$

Note that $\bar{S}_t(f)$ are also the minimum mean-squared estimates of the terms in $\mathbf{S}^m$. The above update rules thus also implicitly impute the missing values of the data.

In some situations the multinomial bases $P(f|z)$ may be available, for instance when they have been learned separately from regions of the data that have no missing time-frequency components. In such situations, only the mixture weights $P_t(z)$ need to be learned for any incomplete spectral vector S_t in order to estimate $\bar{S}_t(f)$. This can be achieved simply by iterations of Equations 10, 11, 12 and 13.

The likelihood of the model for the observed data is guaranteed to converge monotonically, and that has been validated with multiple experiments. The model likelihood over both the observed and the imputed data is not guaranteed to increase all the time. However, after the first few iterations and once the imputation becomes increasingly plausible, the model likelihood always converges and does so monotonically. Figure 2 shows the model's likelihood convergence trajectory for the experiment in the next section.

5 Missing Phase Values

So far we only considered recovering the magnitude spectrum values of spectrograms. As mentioned before these are the values that are most linked to the perceived restoration of the missing data input. The phase values are not so much linked to the audible content of a sound, but rather help describe transient effects and subtle timing information. Because of that, recovering phase values is not that involved a process as long as the resulting reconstruction results in a largely smooth time signal regardless of its content. Estimation of the missing phase is easily done using a straightforward modification of the Griffin-Lim algorithm [18]. This is an iterative process which is defined as follows:

1. Set the phase of the missing entries to random values
2. Transform the resulting spectrogram to the time domain
3. Transform the resulting waveform back to the time-frequency domain and keep the phase values corresponding to the missing entries

4. Use these values as the phase for the missing entries, go to step 2 and repeat until convergence

Throughout this process we keep the magnitude values as well as the known phase values fixed. This algorithm usually converges to a satisfactory solution in 10-20 iterations. For convergence analysis and required conditions on the data refer to [18]. Alternatively we can use the more modern approach in [19], which produces in improved results with fewer constraints.

6 Experimental Evaluation

In this section we will evaluate the algorithms of Section 4 on several examples of spectrograms with missing time-frequency regions, showing both their convergence and effectiveness at imputation. In our examples the spectrograms are from complex musical recordings with multiple, additive concurrent spectral patterns. Such data are particularly difficult to impute using conventional algorithms.

6.1 Illustrative example

We first evaluate our proposed approach with an illustrative example. The input data in this case consisted of a synthetic piano recording of some isolated notes and subsequently of a mixture of these notes. We removed a triangular time-frequency section of the part where the multiple notes took place. We trained our model using the isolated note sections and using the proposed model we learned 60 multinomial bases which we then used to impute the missing values. Figure 3 shows both the original spectrogram (with the missing region marked by the dotted line) and the reconstructed spectrogram. As a comparator we also show the complete spectrograms obtained by imputation of the missing regions using SVD and K-nearest neighbors.

We note that even in this simple problem the K-NN approach does a poor job of modeling all three notes and instead averages out the imputed data thus creating a visibly and audibly incorrect reconstruction. The SVD imputation is more successful due to its additive nature, appropriately borrowing elements to reconstruct the coinciding notes. However this approach cannot guarantee that the imputed output will be non-negative (as required for magnitude spectra) and can potentially return negative values which must be either set to zero, or be rectified, resulting in musical noise. The proposed method properly layers elements of multiple notes to impute the missing data and does not suffer from producing negative values. The result is almost indistinguishable from the ground truth.

Although our approach is not as fast as the k-nearest neighbors approach, it is an order of magnitude faster than the SVD approach since each iteration involves the evaluation of a small number of inner products as opposed to the computation of an entire SVD. The computation times for the above example were 0.7 seconds for k-nearest neighbors, 90 seconds for the SVD and 8 seconds for our proposed approach. Simulations were run on an ordinary laptop computer.

6.2 Real-world examples

In this section we will consider some complex cases derived out of real-world recordings with challenging missing data cases. We note that in these cases strict quantitative evaluation of the results is meaningless. This approach produces results which sound as conformant as possible given an input, but are not guaranteed to approximate the original data. This algorithms effectively "hallucinates" an answer, as opposed to actually recovering true data values. Therefore, any automatic quantitative assessment scores the resulting outputs as fairly dissimilar to the ground truth. Interested readers can listen to results from this paper at `http://www.media.mit.edu/~paris/vlsi10/`.

6.2.1 Coherent Missing Data

We will first examine a more complex case of the above example where we attempt to fill a large continuous gap in the spectrogram for a complex musical piece. This case is meant to exemplify the case where a user might manually mask a section of a spectrogram. The section with the gap was a five second real piano recording of Bach's three-part piano invention #3 in D-BWV 879 [20]. The size of the gap was 4.3 seconds by 3 kHz at its widest extent. We also used 10 seconds of data from another piano piece (two-part invention #3 in D -BWV 774) which provided the needed information to impute the missing data. We extracted 60 multinomial bases while training on both the complete and the incomplete data. As a comparator, we also reconstructed the spectrogram using a rank-60 SVD. The sampling rate was 14700Hz and the spectrum size was 1024 points The imputation results are shown in figure 4. One can find some visual inconsistencies using the SVD most noticeably in the rougher texture of the reconstructed area. In contrast our approach results into visually more plausible results. We also invert the spectrograms back to the time domain in order to perform a listening evaluation. In order to accurately compare the artifacts caused by the imputation we used the phase values from the original signal. The proposed method resulted in virtually inaudible reconstruction artifacts whereas the SVD approach introduced harmonic artifacts which sounded like distorted piano notes as a background noise.

6.2.2 Bandwidth Expansion

In the next example we attempt to perform *bandwidth expansion*. Audio signals can often be bandlimited by having been passed through a restrictive channel, such as a telephone. The spectral excision here is very systematic and consistent. We learn a set of 120 multinomial bases from other wideband training data, and use these to infer the missing spectral content of the bandlimited signal. This presumes that the training data is similar in nature to the testing data (i.e. if we need to resample piano music we should train on piano music). An example of this is shown in figure 5. We removed 80% of the upper frequencies of a ten second rock recording of the song "Back in Black" by the band AC/DC and we trained on an eight second recording by the same band playing a different song ("Highway to Hell"). The sampling rate was once again 14700Hz and the spectrum size was 1024 points. As a comparator we have also reconstructed the spectrogram using rank-120 SVD. The SVD clearly underperforms in both the audible and the visual reconstruction in this experiment, exhibiting an audible spectral smearing in the high frequency registers which dominated

the reconstruction. Our proposed method results in a plausible, although as expected non-exact, reconstruction.

6.2.3 Scattered Missing Data

Here we present a very challenging case where the missing data was evenly and randomly distributed across the input. For this test a smoothed random binary mask was applied to an input sound so that about 60% of the data was removed. The input sound was sampled as 22050Hz and the time-frequency values were obtained using a 1024 point short-time Fourier transform. We modeled the data with a mixture of 60 multinomials. As a comparator, a rank-60 SVD was also used to reconstruct the spectrogram. The results of this experiment are shown in figure 6.

We note that the SVD model results in grainier reconstruction in the missing parts which results in an audibly "muddy" mix as an output. The proposed model results in a crisper reconstruction in which the present musical instruments are much more distinct. Any processing artifacts are virtually imperceptible unless the sound is carefully compared to the original input.

6.2.4 Compression Example

Finally we examine the case of aggressive audio compression. In this example we use a classical piece of music and apply on it an mp3 compression algorithm with a bitrate of 8kbps. This is a very dramatic compression operation which results in loss of data and excessive musical noise. The compression operation removes time-frequency energy from both the higher frequencies, but also from weaker areas in the low frequency range. In order to recover as much missing data as possible we used a clean classical music recording from which to learn a time-frequency model and combined that with a model learned from the available data of the compressed input. The results of this operation are shown in figure 7. Our proposed approach is more successful in suppressing musical noise artifacts and extrapolates a plausible upper frequency reconstruction. In contrast the SVD provides a poor reconstruction of the upper frequencies and exhibits more distortion.

7 Conclusions

In this paper we presented a data imputation approach that is best suited for non-negative data, and presented its application in restoring sounds with missing time-frequency components. We showed how interpreting time-frequency distributions as histograms or counts data, we can decompose them in a manner which is more appropriate than generic techniques based on the SVD and K-NN methods. We demonstrated the performance of this approach using a variety of problems inspired from real-world situations and have shown that it performs better than generic missing data approaches. The model we have shown is ignoring the temporal dimension, and is instead using any existing frequency information in order to impute the missing values. This creates a problem when all frequencies are missing simultaneously, therefore this approach is ill-suited for problems that involve filling gaps of complete silence. As shown by [5], temporal models can be used to address this issue, and future work on this project

can concentrate on convolutive formulations of the proposed algorithm which should be able to address this particular situation.

References

1. Raj, B. Reconstruction of Incomplete Spectrograms for Robust Speech Recognition, Ph.D. Dissertation, Carnegie Mellon University, May 2000.
2. Roweis, S.T. One Microphone Source Separation. NIPS 2000: 793-799.
3. Brand, M. E. Incremental Singular Value Decomposition of Uncertain Data with Missing Values, European Conference on Computer Vision (ECCV), Vol 2350, pps 707-720, May 2002.
4. Reyes-Gomez, M.J., N. Jojic and D.P.W. Ellis. Detailed graphical models for source separation and missing data interpolation in audio. 2004 Snowbird Learning Workshop Snowbird, Utah
5. Le Roux, J, H. Kameoka, N. Ono, A. de Cheveigné and S. Sagayama. Computational Auditory Induction by Missing-Data Non-Negative Matrix Factorization, SAPA 2008, Brisbane, Australia.
6. Shashanka, M., B. Raj, P. Smaragdis. Sparse Overcomplete Latent Variable Decomposition of Counts Data. NIPS 2000.
7. Smith, J. O. Spectral Audio Signal Processing, March 2007 Draft, `http://ccrma.stanford.edu/~jos/sasp/`, accessed June 2008.
8. David, M. H., Little, R. J. A., Samuhel, M. E. and Triest, R. K. (1983). Imputation methods based on the propensity to respond, Proceedings of the Business and Economics Section, American Statistical Association.
9. Quinlan, J. R. (1989). Unknown attribute values in induction, Proc. of the Sixth International Conference on Machine Learning
10. Ghaharamani, Z. and Jordan, M. I. (1994). Learning from incomplete data. Technical report AI Memo 1509, Artificial Intelligence Laboratory, MIT
11. Raj, B., M. L. Seltzer, and R. M. Stern, Reconstruction of Missing Features for Robust Speech Recognition, Speech Communication Journal 43(4): 275-296, September 2004
12. Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P. and Botstein, D. Imputing Missing Data for Gene Expression Arrays. Technical report (1999), Stanford Statistics Department.
13. Hofmann, T. Learning the Similarity of Documents : an information-geometric approach to document retrieval and categorization, Advances in Neural Information Processing Systems 12, pp-914-920, MIT Press, 2000
14. Hofmann, T. and J. Puzicha. Unsupervised learning from dyadic data. TR 98-042, ICSI, Berkeley, CA, 1998.
15. Hazewinkel, M. Encyclopedia of Mathematics. http://eom.springer.de/
16. http://en.wikipedia.org/wiki/Negative_binomial_distribution
17. Dempster, A.P., N.M. Laird, D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B, 39, 1-38. 1977.
18. Griffin D.W. and J.S. Lim. Signal reconstruction from short-time Fourier transform magnitude, in the IEEE Transactions of Acoustics, Speech, and Signal Processing, 32(2):236-243, 1984.
19. Bouvrie, J. and T. Ezzat. An Incremental Algorithm for Signal Reconstruction from Short-Time Fourier Transform Magnitude, in Interspeech 2006, Pittsburgh, USA.
20. Gould, G. Bach: The Two and Three Part Inventions - The Glenn Gould Edition, by SONY classics, ASIN B000GF2YZ8, 1994.

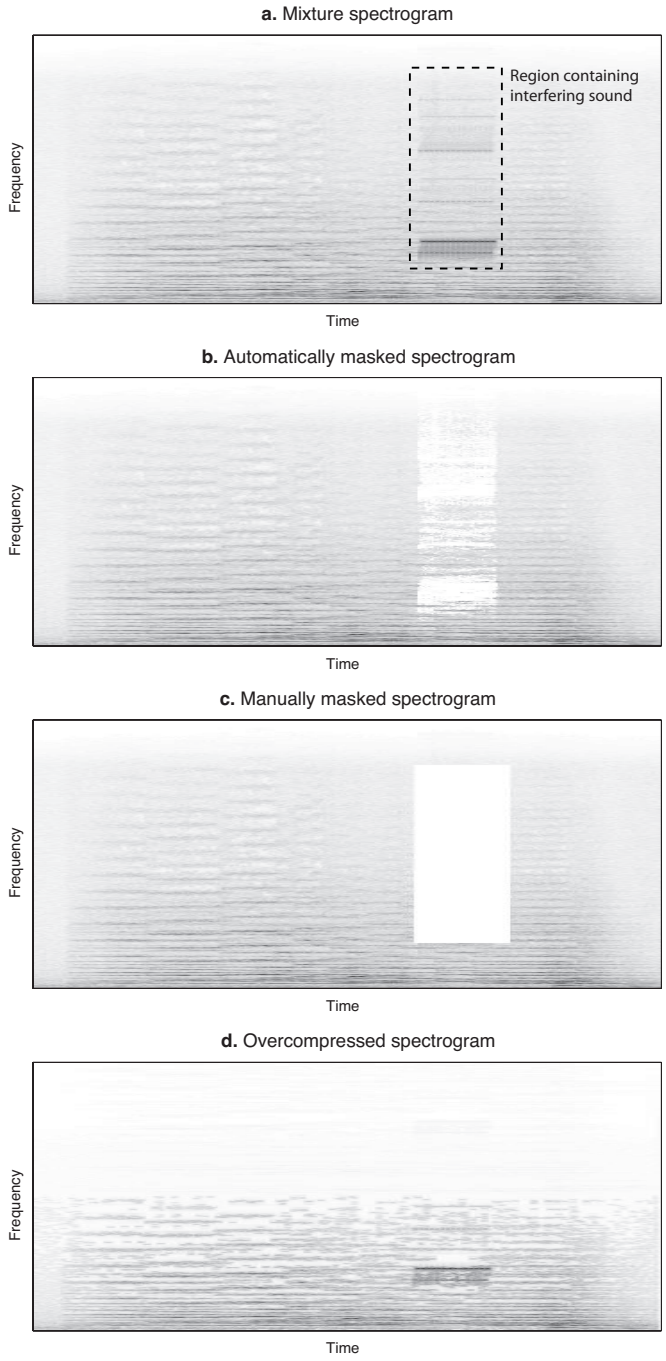

Fig. 1 **a**. Magnitude spectrogram of a mixture of classical music and a phone ring. The spectrogram visualizes the spectral patterns for both signals – the long parallel lines represent the harmonic elements present in the music signal, whereas the phone ring is represented by the rectangular clusters in the outlined region. **b**. In this spectrogram time-frequency components not belonging to the music signal have been removed by a source separation algorithm. **c**. Here time-frequency regions dominated by the phone ring have been manually edited out by a user. **d**. In this example overzealous compression has removed a significant part of the time-frequency content.

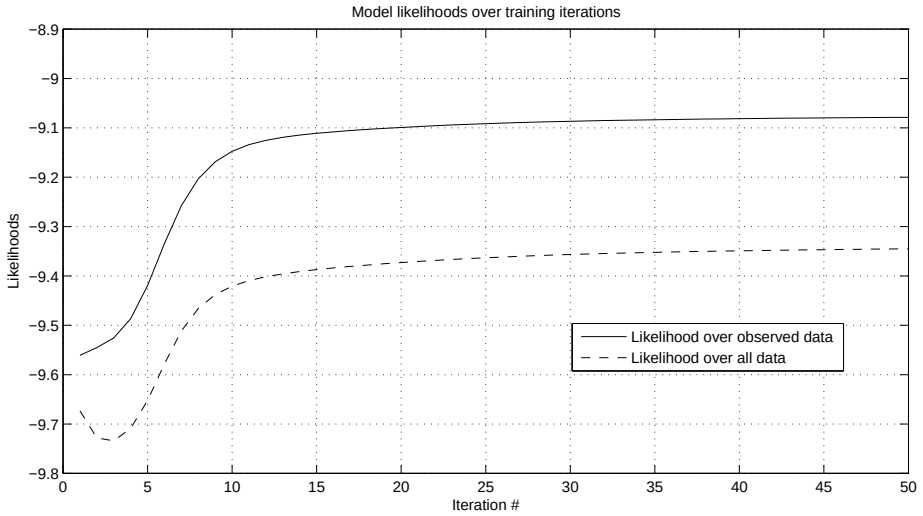

Fig. 2 Likelihood of observed data $\mathbf{S}^o$ (solid line) and complete spectrogram $\mathbf{S}$ (dashed line) as a function of iteration for the data in Figure 6.

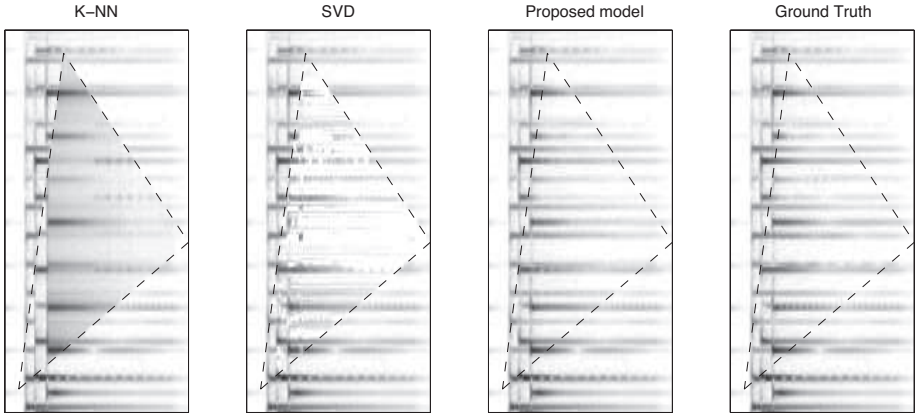

Fig. 3 Comparison of three data imputation approaches on a simple problem. The missing data area is denoted by the dashed black line in all plots. The first plot from the left shows the results from K-NN imputation, the second from SVD imputation, the third using our proposed model and the fourth is the ground truth.

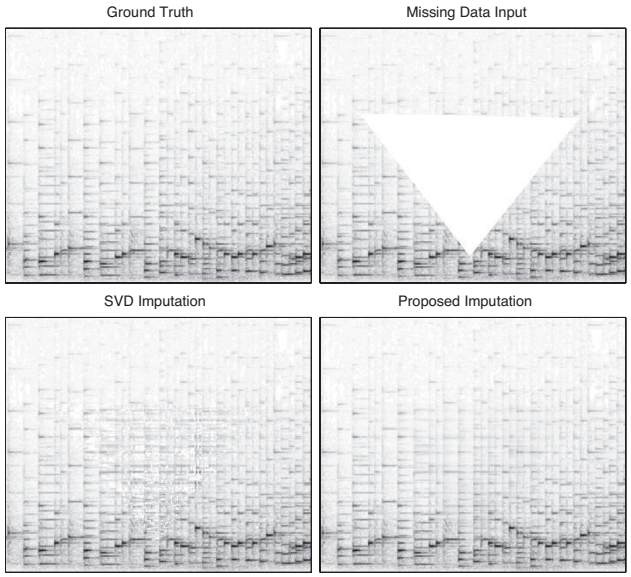

Fig. 4 Example reconstruction of a gap filling experiment. The leftmost plot shows the actual data, the second plot shows the input with the large gap removing about 15% of the data, we have zoomed into the region of the gap by not plotting some of the higher frequency content. the third plot is the SVD reconstruction and the fourth plot is our proposed method.

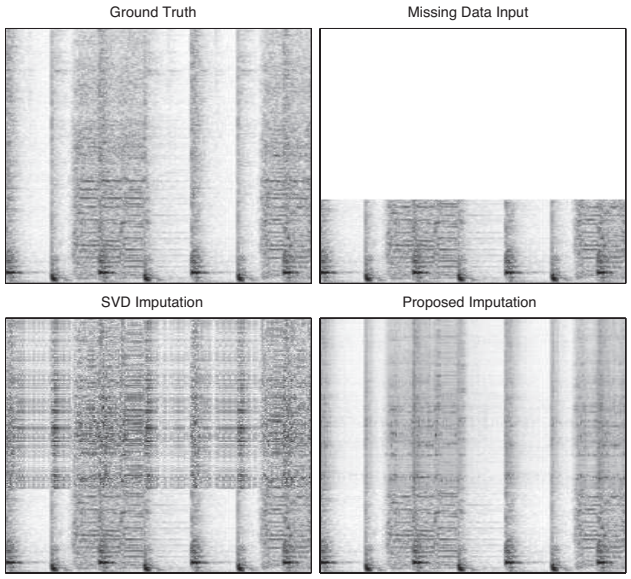

Fig. 5 Example reconstruction of a bandwidth expansion. In the leftmost plot the original signal is shown. The second plot displays the bandlimited input we used where 80% of the top frequencies were removed. The third plot is the SVD reconstruction and the fourth plot is the reconstruction using our model.

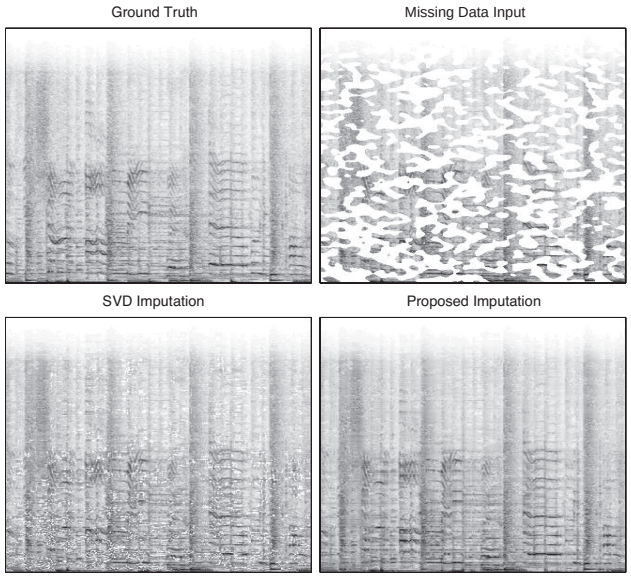

Fig. 6 Example reconstruction of a music signal with a binary mask occluding roughly 60% of the samples. The leftmost plot shows the original signal, the second plot shows the masked input we used for the reconstruction, the third plot shows the reconstruction using the SVD and the fourth one shows the reconstruction using our model.

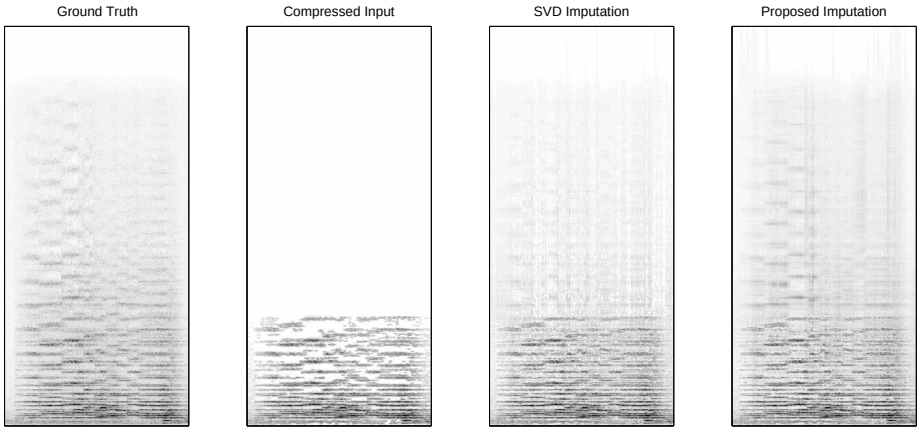

Fig. 7 Results of recovering overly compressed audio data. The leftmost panel displays the original input signal. The second panel displays the signal after it was compressed at a bitrate of 8kbps. Note how the highest frequencies, as well as areas of lesser energy have been removed by the compression algorithm. The third panel displays the reconstruction using an SVD approach and the fourth panel displays the results of our proposed method. Note how our results have a better defined high frequency range and are slightly less noisy in the low frequencies.