Transcription and classification of audio data by sparse representations and geometric methods

Michal Genussov

Department of Electrical Engineering

Advisor: Prof. Israel Cohen
1. Background and motivation

2. Transcription of polyphonic music by Sparse Representations

3. Classification of audio and speech data using geometric methods

4. Conclusions
Outline

1. **Background and motivation**
   - Transcription of polyphonic music
   - Classification of audio and speech data

2. Transcription of polyphonic music by Sparse Representations

3. Classification of audio and speech data using geometric methods

4. Conclusions
Polyphonic music

- More than one note is played at a time
Polyphonic music

- More than one note is played at a time
- Can be produced from a single instrument
Polyphonic music

- More than one note is played at a time
- Can be produced from a single instrument or from several instruments
**Definition: Transcription of music**

Identification of musical parameters in an audio signal which are required to write down the score sheet of the notes.
Transcription of music

Definition: Transcription of music

Identification of musical parameters in an audio signal which are required to write down the score sheet of the notes.

- We refer only to the task of identification of notes.
What is it good for?

- Information retrieval and classification
What is it good for?

- Information retrieval and classification
- A helpful tool for musicians
What is it good for?

- Information retrieval and classification
- A helpful tool for musicians
- Modifying, rearranging and processing music
What is it good for?

- Information retrieval and classification
- A helpful tool for musicians
- Modifying, rearranging and processing music
- Compression - WAV to MIDI transformation
What is it good for?

- Information retrieval and classification
- A helpful tool for musicians
- Modifying, rearranging and processing music
- Compression - WAV to MIDI transformation
- Interactive music systems
Classification of audio and speech data

Examples:
- Musical genres (Jazz, Rock, Classical etc.)
- Musical instruments
- Audio mood identification
- Speaker identification
- Phonemes identification
Examples:

- Musical genres (Jazz, Rock, Classical etc.)
- Musical instruments
- Audio mood identification
- Speaker identification
- Phonemes identification
- and many more...

MIREX annual competition
(http://www.music-ir.org/mirex/wiki/2010:Audio_Classification_(Train/Test)_Tasks)
What is it good for?

- Structure and organize large data bases (on the web, for radio stations and more)
What is it good for?

- Structure and organize large data bases (on the web, for radio stations and more)
- Similarity retrieval
What is it good for?

- Structure and organize large data bases (on the web, for radio stations and more)
- Similarity retrieval
- Audio thumbnailing
What is it good for?

- Structure and organize large data bases (on the web, for radio stations and more)
- Similarity retrieval
- Audio thumbnailing
- Smart video conference systems
What is it good for?

- Structure and organize large data bases (on the web, for radio stations and more)
- Similarity retrieval
- Audio thumbnailing
- Smart video conference systems
- Advanced hearing aids
Outline

1. Background and motivation

2. Transcription of polyphonic music by Sparse Representations
   - Transcription of polyphonic music
   - Sparse Representations
   - Dictionary learning
   - Musically structured dictionary learning
   - Experimental results

3. Classification of audio and speech data using geometric methods

4. Conclusions
Harmonic instruments act as periodic oscillators, leading to vibrations in a fundamental frequency (pitch) and in its integer multiples, i.e. its harmonics.
The difficulty in transcription of polyphonic music

In polyphonic music, the harmonics of notes tend to overlap (especially in western music).

The spectrum of C4 of a piano
The difficulty in transcription of polyphonic music

In polyphonic music, the harmonics of notes tend to overlap (especially in western music).

The spectrums of C4 & E4 of a piano
The difficulty in transcription of polyphonic music

In polyphonic music, the harmonics of notes tend to overlap (especially in western music).

The spectrums of C4, E4 & G4 of a piano
The difficulty in transcription of polyphonic music

In polyphonic music, the harmonics of notes tend to overlap (especially in western music).

The spectrums of C4, E4 & G4 of a piano
Previous approaches for transcription

[Marolt, 2004]
An auditory model followed by multiple pitch tracking using adaptive oscillators.
Previous approaches for transcription

[Marolt, 2004]
An auditory model followed by multiple pitch tracking using adaptive oscillators.

[Cemgil et al., 2006]
A Bayesian framework in which both high level (cognitive) prior information is coupled with low level (acoustic physical) information in a principled manner.
Previous approaches for transcription

[Abdallah and Plumbley, 2006]

Transcription of polyphonic music by sparse coding.
Previous approaches for transcription

[Abdallah and Plumbley, 2006]
Transcription of polyphonic music by sparse coding.

[Poliner & Ellis, 2007]
Classification by SVM of frame-level note instances, according to spectral features, followed by hidden Markov model post processing.
Previous approaches for transcription

[Saito et al., 2008]

“Specmurt” analysis of the music, assuming a common distribution of the harmonics intensities.
Previous approaches for transcription

[Saito et al., 2008]
“Specmurt” analysis of the music, assuming a common distribution of the harmonics intensities.

[Klapuri, 2008]
A computational model which simulates the auditory system, followed by an iterative pitch estimation.
Limitations of previous approaches

- Require a diverse and large data set
Limitations of previous approaches

- Require a diverse and large data set
- The assumption of a common distribution of the intensities of the harmonics is inaccurate
Limitations of previous approaches

- Require a diverse and large data set
- The assumption of a common distribution of the intensities of the harmonics is inaccurate
  ⇒ The intensities of the harmonics change unevenly according to
    - The musical instrument
    - The intensity
    - Time
    - The fundamental frequency
Our solution

We offer an algorithm which

- Learns the intensities of the harmonics from the signal itself.

![The spectrum of C4 of a piano](image)
Our solution

We offer an algorithm which

- Learns the intensities of the harmonics from the signal itself.
- Is based on the common structure of music spectra, thus does not require a diverse or large data set.

The spectrum of C4 of a piano
Our solution

We offer an algorithm which

- Learns the intensities of the harmonics from the signal itself.
- Is based on the common structure of music spectra, thus does not require a diverse or large data set.
- Is based on sparse representations with a suitable parametric dictionary.

The spectrum of C4 of a piano
Sparse coding

Sparse coding is formulated as solving the problem

\[ P_0 : \min_x \|x\|_0 \quad \text{s.t.} \quad Ax = y \]  

(1)
Sparse coding is formulated as solving the problem

\[ P_0 : \min_x \|x\|_0 \quad \text{s.t.} \quad Ax = y \]  \hspace{1cm} (1)

In an approximated formulation

\[ \min_x \|Ax - y\|_2^2 \quad \text{s.t.} \quad \|x\|_0 \leq K \]  \hspace{1cm} (2)
Sparse coding is formulated as solving the problem

\[ P_0 : \min_x \|x\|_0 \quad \text{s.t.} \quad Ax = y \] (1)

In an approximated formulation

\[ \min_x \|Ax - y\|_2^2 \quad \text{s.t.} \quad \|x\|_0 \leq K \] (2)

- The sparse coding problem is NP-hard
  \[ \Rightarrow \text{Approximation using a pursuit algorithm.} \]
Sparse coding of music

- The number of notes played at a time is limited \((K)\)
The number of notes played at a time is limited ($K$) 
$\Rightarrow$ Transcription of music can be interpreted as a sparse coding problem.

Sparse representation of music in the frequency domain is convenient and useful.
Several types of pursuit algorithms:

Pursuit algorithms for solving the sparse coding

Several types of pursuit algorithms:


- **Relaxation methods** - Replace the $\ell_0$ “norm” by
  - Other $\ell_p$ norms for some $p \in (0, 1]$ - FOCUSS, Basis Pursuit
  - Smooth functions such as
    \[ \sum_{i=1}^{m} 1 - e^{-\alpha x_i^2} \text{ or } \sum_{i=1}^{m} \frac{x_i^2}{(\alpha + x_i^2)} \]

![Diagram showing the value of $x_i$ in $\ell_0$ norm and its smooth version.](image-url)
Choosing a dictionary

There are several possibilities for choosing a dictionary:

1. Analytic dictionaries
2. Learned dictionaries
3. Parametric dictionaries
The dictionary $A$ is pre-defined and constant.

Examples for transforms under which images were found to be sparse:

- Curvelets [Candes, 2000]
- Contourlets [Do & Vetterli, 2005]
- Wavelets [Mallat, 1993]
- Short-time Fourier transform
The dictionary $A$ is pre-defined and constant.

Examples for transforms under which images were found to be sparse:

- Curvelets [Candes, 2000]
- Contourlets [Do & Vetterli, 2005]
- Wavelets [Mallat, 1993]
- Short-time Fourier transform

⇒ Very fast, but don’t represent the data very well
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- The original optimization problem

$$\min_{x} \|A x - y\|^2_2 \quad \text{s.t.} \quad \|x\|_0 \leq K$$
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- For multiple signals

\[
\min_X \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i
\]
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

$$\min_{A,X} \|AX - Y\|^2_F \text{ s.t. } \|x_i\|_0 \leq K_i$$
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

\[
\min_{A,X} \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i
\]

\[\text{Y} \quad = \quad \text{A} \quad \times \quad \text{X}\]
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

$$\min_{A, X} \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i$$
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

\[
\min_{A,X} \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i
\]
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

\[
\min_{A,X} \|AX - Y\|_F^2 \quad \text{s.t. } \|x_i\|_0 \leq K_i
\]
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

\[
\min_{A, X} \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i
\]
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

\[
\min_{A,X} \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i
\]
Learned dictionaries [Olshausen & Field, 1996]

- The dictionary is explicit and learned during the process.
- When learning the dictionary

$$\min_{A,X} \|AX - Y\|_F^2 \quad \text{s.t.} \quad \|x_i\|_0 \leq K_i$$

\[\begin{array}{ccc}
Y & = & X
\end{array}\]
Learned dictionaries

Two main algorithms for dictionary learning:

- Method of Optimal Directions (MOD) [Engan et al., 2000]
- K-SVD [Aharon et al., 2006]

Both based on 2 stages operating alternately:

1. Sparse coding
2. Dictionary update
Learned dictionaries

Two main algorithms for dictionary learning:

- Method of Optimal Directions (MOD) [Engan et al., 2000]
- K-SVD [Aharon et al., 2006]

Both based on 2 stages operating alternately:

1. Sparse coding
2. Dictionary update

\[\Rightarrow\] Data-driven, but slow and require a large data set.
Method of Optimal Directions (MOD) [Engan et al., 2000]
Method of Optimal Directions (MOD) [Engan et al., 2000]

Initialize dictionary either by using random entries, or by using $m$ randomly chosen samples
Method of Optimal Directions (MOD) [Engan et al., 2000]

Initialize dictionary either by using random entries, or by using \( m \) randomly chosen samples

Sparse coding by a pursuit algorithm
Method of Optimal Directions (MOD) [Engan et al., 2000]

Initialize dictionary either by using random entries, or by using \( m \) randomly chosen samples

Sparse coding by a pursuit algorithm

Dictionary update by least squares:

\[
A_{(k)} = \operatorname{argmin}_A \|Y - AX_{(k)}\|_F^2 = YX^\dagger_{(k)}
\]
Method of Optimal Directions (MOD) [Engan et al., 2000]

Initialize dictionary either by using random entries, or by using $m$ randomly chosen samples.

Sparse coding by a pursuit algorithm

Dictionary update by least squares:

$$A^{(k)} = \arg \min_A \| Y - AX^{(k)} \|_F^2 = YX^\dagger^{(k)}$$
Method of Optimal Directions (MOD) [Engan et al., 2000]

- Initiate the dictionary either by using random entries, or by using $m$ randomly chosen samples.

- Sparse coding by a pursuit algorithm.

- Dictionary update by least squares:
  \[ A_{(k)} = \arg\min_A \| Y - AX_{(k)} \|_F^2 = YX^\dagger_{(k)} \]

- Stopping criterion: if the change in $\| Y - A_{(k)}X_{(k)} \|_F^2$ is small enough, stop.
K-SVD [Aharon et al., 2006]
K-SVD [Aharon et al., 2006]

1. Initialize dictionary: either by using random entries, or by using $m$ randomly chosen samples.

2. Sparse coding by a pursuit algorithm.

3. K-SVD Dictionary update:
   - For each atom separately.

Stopping criterion: if the change in $\|Y - A(k)X(k)\|_F^2$ is small enough, stop.
**K-SVD** [Aharon et al., 2006]

- **Initialize dictionary** either by using random entries, or by using $m$ randomly chosen samples.

- **Sparse coding** by a pursuit algorithm.

- **K-SVD Dictionary update** - *For each atom separately*.

- **Stopping criterion**: if the change in $\|Y - A(k)X(k)\|_F$ is small enough, stop.
K-SVD [Aharon et al., 2006]

K-SVD Dictionary update
For the atom $a_{j0}$:
K-SVD [Aharon et al., 2006]

K-SVD Dictionary update
For the atom $a_{j_0}$:

Define the error matrix

$$E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T$$
**K-SVD** [Aharon et al., 2006]

**K-SVD Dictionary update**
For the atom $a_{j_0}$:

Define the error matrix
$$E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T$$

Restrict $E_{j_0}$ only to the columns corresponding to the signals in $Y$ that use $a_{j_0}$ and get $E_{j_0}^R$
K-SVD [Aharon et al., 2006]

**K-SVD Dictionary update**
For the atom $a_{j_0}$:

Define the error matrix

$$E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T$$

Restrict $E_{j_0}$ only to the columns corresponding to the signals in $Y$ that use $a_{j_0}$ and get $E_{j_0}^R$

Apply SVD decomposition of the matrix $E_{j_0}^R$
Transcription of polyphonic music
Sparse Representations
Dictionary learning
Musically structured dictionary learning
Experimental results

K-SVD [Aharon et al., 2006]

K-SVD Dictionary update
For the atom \( a_{j_0} \):

Define the error matrix
\[
E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T
\]

Restrict \( E_{j_0} \) only to the columns corresponding to the signals in \( Y \) that use \( a_{j_0} \) and get \( E_{j_0}^R \)

Update the atom \( a_{j_0} \) and the coefficients \( x_{j_0}^T \) according to rank-1 approximation

Apply SVD decomposition of the matrix \( E_{j_0}^R \)
Parametric dictionaries

- A compromise between analytic and learned dictionaries
- Structured dictionaries, in which a limited set of parameters are learned
Parametric dictionaries

- A compromise between analytic and learned dictionaries
- Structured dictionaries, in which a limited set of parameters are learned

Examples:
- The double sparsity model [Rubinstein, Zibulevsky & Elad, 2010]
- The image signature dictionary [Aharon & Elad, 2008]
A parametric dictionary for music spectra

- Musical tones share a common structure of their spectra:

![FFT of the note A4](image-url)
A parametric dictionary for music spectra

- Musical tones share a common structure of their spectra:
  - The FFT of a musical tone can be modeled as an impulse train which was convolved by a shaping filter.
  - The shaping filter - represents the timbre of the sound (its “color”).

[Graph of FFT of note A4]
The Constant Q transform (CQT)

- The Short-time Fourier transform (STFT) - a series of linearly spaced filters.

\[
\delta f_k = \text{const}
\]
The Constant Q transform (CQFT)

- The Short-time Fourier transform (STFT) - a series of linearly spaced filters.

\[ \delta f_k = \text{const} \]

- The Constant Q transform (CQT) - a series of logarithmically spaced filters, thus suitable for the human hearing.

\[ \delta f_k = 2^{\frac{1}{n}} \delta f_{k-1} \]

- \( n \) = the number of filters per octave

- \( Q = \frac{f_k}{\delta f_k} \)
The Musically-Structured (MS) dictionary [Genussov & Cohen, 2010a]

- We define a dictionary where each atom corresponds to the CQT of a different note in the piano (total 88 atoms).
The Musically-Structured (MS) dictionary [Genussov & Cohen, 2010a]

- We define a dictionary where each atom corresponds to the CQT of a different note in the piano (total 88 atoms).
  - The location of the support (non zero elements) is constant.
  - The entries in the support are learned.
The Musically-Structured (MS) dictionary [Genussov & Cohen, 2010a]

- We define a dictionary where each atom corresponds to the CQT of a different note in the piano (total 88 atoms).
  - The location of the support (non zero elements) is constant.
  - The entries in the support are learned.
Advantages of the MS dictionary

Better representation of the signal

- The support of the dictionary is constant and represents all the notes in the piano
  ⇒ Doesn’t require a large and diverse data set

- The values of the entries in the support are learned
  ⇒ Adjustment to the *timbre* of the signal
Advantages of the MS dictionary

The complexity of the dictionary is proportional to its learned parameters

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Explicit dictionary</th>
<th>Musically structured dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n \cdot m )</td>
<td>( h \cdot m )</td>
</tr>
</tbody>
</table>

\( n \) is the number of CQT bins, \( m \) is the number of atoms and \( h \) is the number of harmonics

\( h < m \Rightarrow \) Our dictionary is more computationally efficient than an explicit dictionary
**Musically-Structured dictionary learning**

**Initialize dictionary** by the evaluated CQT of the notes in the piano

**Sparse coding** by a pursuit algorithm

**Dictionary update:** Update only the elements in the support of \( \mathbf{A} \)

**Stopping criterion:** if the change in \( \| \mathbf{Y} - \mathbf{A}_{(k)} \mathbf{X}_{(k)} \|_F^2 \) is small enough, stop.
Musically-Structured MOD (MS-MOD) [Genussov & Cohen, 2010a]
Musically-Structured MOD (MS-MOD) [Genussov & Cohen, 2010a]

- Initialize dictionary by the evaluated CQT of the notes in the piano

  Sparse coding by a pursuit algorithm

  MS-MOD Dictionary update

- Stopping criterion: if the change in $\|Y - A_{(k)}X_{(k)}\|_F^2$ is small enough, stop.
Musically-Structured MOD (MS-MOD) [Genussov & Cohen, 2010a]

Define the dictionary of used atoms $\tilde{A}$ and the corresponding matrix $\tilde{X}$

MS-MOD Dictionary update:
Musically-Structured MOD (MS-MOD) [Genussov & Cohen, 2010a]

**MS-MOD Dictionary update:**

Define the dictionary of used atoms $\tilde{A}$ and the corresponding matrix $\tilde{X}$

Least squares:

$$\tilde{A}_{(k)} = \arg\min_{\tilde{A}} ||Y - \tilde{A}\tilde{X}_{(k)}||_F^2 = Y\tilde{X}_{(k)}^\dagger$$
Musically-Structured MOD (MS-MOD) [Genussov & Cohen, 2010a]

MS-MOD Dictionary update:

Define the dictionary of used atoms $\tilde{A}$ and the corresponding matrix $\tilde{X}$

Least squares:

$$\tilde{A}_{(k)} = \arg\min_{\tilde{A}} \| Y - \tilde{A} \tilde{X}_{(k)} \|_F^2 = Y \tilde{X}_{(k)}^\dagger$$

Zero the entries out of the support of $\tilde{A}$
Musically-Structured MOD (MS-MOD) [Genussov & Cohen, 2010a]

MS-MOD Dictionary update:

Define the dictionary of used atoms $\tilde{A}$ and the corresponding matrix $\tilde{X}$

Add the unused atoms to get the updated $A$

Least squares:

$$\tilde{A}_{(k)} = \arg\min_{\tilde{A}} \| Y - \tilde{A}\tilde{X}_{(k)} \|^2_F = Y\tilde{X}_{(k)}^\dagger$$

Zero the entries out of the support of $\tilde{A}$
Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

1. **Initialize dictionary** by the evaluated CQT of the notes in the piano.

2. **Sparse coding** by a pursuit algorithm.

3. **MS-K-SVD Dictionary update**.

**Stopping criterion**: if the change in $\| Y - A_{(k)} X_{(k)} \|_F^2$ is small enough, stop.
Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

**MS-K-SVD Dictionary update**
For the used atom $a_{j0}$:
Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

**MS-K-SVD Dictionary update**

For the used atom $a_{j_0}$:

Define the error matrix

$$E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T$$
Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

MS-K-SVD Dictionary update
For the used atom $\mathbf{a}_{j_0}$:

Define the error matrix
$$\mathbf{E}_{j_0} = \mathbf{Y} - \sum_{j \neq j_0} \mathbf{a}_j \mathbf{x}_j^T$$

Restrict the columns as in K-SVD to get $\mathbf{E}^R_{j_0}$, and restrict the rows according to the support to get $\tilde{E}^R_{j_0}$.
Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

MS-K-SVD Dictionary update
For the used atom $a_{j_0}$:

Define the error matrix
$$E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T$$

Restrict the columns as in K-SVD to get $E_{j_0}^{R}$, and restrict the rows according to the support to get $\hat{E}_{j_0}^{R}$

Apply SVD decomposition of the matrix $\hat{E}_{j_0}^{R}$
Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

**MS-K-SVD Dictionary update**
For the used atom $a_{j_0}$:

1. Define the error matrix
   \[
   E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T
   \]

2. Restric the columns as in K-SVD to get $E_{j_0}^R$, and restrict the rows according to the support to get $\tilde{E}_{j_0}^R$

3. Update the atom $\tilde{a}_{j_0}$ and the coefficients $\tilde{x}_{j_0}^T$ according to rank-1 approximation

4. Apply SVD decomposition of the matrix $\tilde{E}_{j_0}^R$
The musically-structured algorithm for transcription
The musically-structured algorithm for transcription

Onset detection
The musically-structured algorithm for transcription

- Onset detection
- Wait 32 ms
- Evaluate the number of notes (K)
The musically-structured algorithm for transcription

1. Onset detection
2. Wait 32 ms
3. Evaluate the number of notes (K)
4. Apply CQT
The musically-structured algorithm for transcription

1. **Onset detection**
2. Wait 32 ms
3. Evaluate the number of notes (K)
4. Apply CQT
5. Transcription using MS-MOD or MS-KSVD
Synthesized MIDI music

- Monophonic music

Accuracy = 100%

Accuracy = 100%
Synthesized MIDI music

- Monophonic music

**Definition [Dixon, 2000]**

\[
\text{Accuracy} = \frac{TP}{TP + FP + FN}
\]

TP = True Positive, FP = False Positive, FN = False Negative.

Accuracy = 100%
Synthesized MIDI music

- Simple polyphonic music

![Diagram of piano roll and identified piano rolls with accuracy](image)

Accuracy = 69.6%

Accuracy = 67.7%
Synthesized MIDI music

- Simple polyphonic music
  - The piano roll
  - The identified piano roll - MS-MOD
  - The identified piano roll - MS-K-SVD
  - Accuracy = 69.6%
  - Accuracy = 67.7%

- Complicated polyphonic music
  - The piano roll
  - The identified piano roll - MS-MOD
  - The identified piano roll - MS-K-SVD
  - Accuracy = 64.0%
  - Accuracy = 64.5%
In order to evaluate the results of MOD and KSVD, we used Harmonic product spectrum [Noll, 1969] to assign each atom to a note.

<table>
<thead>
<tr>
<th></th>
<th>MOD</th>
<th>KSVD</th>
<th>MS-MOD</th>
<th>MS-K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monophonic music</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Simple polyphonic music</td>
<td>37.6</td>
<td>39.5</td>
<td>69.6</td>
<td>67.7</td>
</tr>
<tr>
<td>Complicated polyphonic music</td>
<td>43.5</td>
<td>42.7</td>
<td>64.0</td>
<td>64.5</td>
</tr>
</tbody>
</table>
Synthesized MIDI music - a comparison to using an explicit dictionary

- Accords detection - MS-MOD and MS-K-SVD
Synthesized MIDI music - a comparison to using an explicit dictionary

- Accords detection - MS-MOD and MS-K-SVD

- Accords detection - MOD and K-SVD

- black = true positive, red = false positive, yellow = false negative.
Recorded piano music - a comparison to using an explicit dictionary

- Monophonic detection - MS-MOD and MS-K-SVD
Recorded piano music - a comparison to using an explicit dictionary

- Monophonic detection - MS-MOD and MS-K-SVD

- Monophonic detection - MOD and K-SVD
Recorded piano music - a comparison to using an explicit dictionary

Accords detection - MS-MOD and MS-K-SVD

Accuracy = 54.8%

Accuracy = 50.0%
Recorded piano music - a comparison to using an explicit dictionary

- **Accords detection - MS-MOD and MS-K-SVD**
  - Accuracy = 54.8%
  - Accuracy = 50.0%

- **Accords detection - MOD and K-SVD**
  - Accuracy = 14.3%
  - Accuracy = 14.3%
Outline

1. Background and motivation

2. Transcription of polyphonic music by Sparse Representations

3. Classification of audio and speech data using geometric methods
   - Description of the problem
   - Manifold learning & Diffusion maps
   - Experimental results

4. Conclusions
Traditional classification methods of audio and speech

1. Feature extraction - temporal and spectral features
   - Mel-Frequency Cepstral Coefficients
   - Zero Crossing Rate
   - Spectral Flux
   - Spectral Centroid
   - Spectral Rolloff
Traditional classification methods of audio and speech

1. Feature extraction - temporal and spectral features
   - Mel-Frequency Cepstral Coefficients
   - Zero Crossing Rate
   - Spectral Flux
   - Spectral Centroid
   - Spectral Rolloff

2. Classification
   - K-nearest neighbors (K-NN)
   - Linear discriminant analysis
   - Quadratic discriminant analysis
   - Support vector machines (SVM)
The limitations of traditional classification methods

1. “The curse of dimensionality” - The sample complexity and the computational complexity tend to grow exponentially with the dimension.
The limitations of traditional classification methods

1. “The curse of dimensionality” - The sample complexity and the computational complexity tend to grow exponentially with the dimension.
2. Not adaptive to the intrinsic geometry of the feature vectors.

Assumption
The feature vectors of natural audio and speech data lie on a non-linear manifold.
The limitations of traditional classification methods

1. “The curse of dimensionality” - The sample complexity and the computational complexity tend to grow exponentially with the dimension.
2. Not adaptive to the intrinsic geometry of the feature vectors.

Assumption

The feature vectors of natural audio and speech data lie on a non-linear manifold.
Manifold learning

- Allows performing non-linear dimensionality reduction unlike principal component analysis (PCA) or independent component analysis (ICA).
- Aims to reveal the meaningful geometric structure of the feature-vectors.
Manifold learning

- Allows performing non-linear dimensionality reduction **unlike** principal component analysis (PCA) or independent component analysis (ICA).
- Aims to reveal the meaningful geometric structure of the feature-vectors.
- Techniques
  - Isomap [Tenenbaum et al., 2000]
  - Locally-linear embedding (LLE) [Rowies & Saul, 2000]
  - Laplacian eigenmaps [Belkin & Niyogi, 2001]
  - Hessian eigenmaps [Donoho & Grimes, 2003]
  - **Diffusion maps** [Coifman & Lafon, 2006]
Manifold learning

Can be used as an intermediate stage in the classification process, which turns into:

1. Feature extraction
2. Manifold learning (Diffusion maps)
3. Classification
1. Build a connected graph from the feature vectors.
   - The nodes of the graph are the feature vectors \( \{x_i\}_i \).
   - The connections between the nodes are weighted by a symmetric, pointwise nonnegative kernel.
Diffusion maps - an overview

1. Build a connected graph from the feature vectors.
   - The nodes of the graph are the feature vectors \( \{x_i\}_i \).
   - The connections between the nodes are weighted by a symmetric, pointwise nonnegative kernel.

2. Define a Markov random walk on the graph and a right stochastic matrix describing its transitions.
1. Build a connected graph from the feature vectors.
   - The nodes of the graph are the feature vectors \( \{x_i\}_i \).
   - The connections between the nodes are weighted by a symmetric, pointwise nonnegative kernel.

2. Define a Markov random walk on the graph and a right stochastic matrix describing its transitions.

3. Define the **Diffusion distances** \( D_t(x_i, x_j) \) as representing the connectivity in the graph.
Diffusion maps - an overview

1. Build a connected graph from the feature vectors.
   - The nodes of the graph are the feature vectors \( \{ x_i \} \).
   - The connections between the nodes are weighted by a symmetric, pointwise nonnegative kernel.

2. Define a Markov random walk on the graph and a right stochastic matrix describing its transitions.

3. Define the **Diffusion distances** \( D_t(x_i, x_j) \) as representing the connectivity in the graph.

4. Apply spectral decomposition of the right stochastic matrix, and derive the **Diffusion map** \( \Psi_t(x_i) \) of the feature vector \( x_i \).
Proposition

The diffusion map embeds the feature vectors into a lower-dimensional Euclidean space, so that in this space, the Euclidean distance approximates the diffusion distance, or equivalently

\[ D_t(x_i, x_j) \approx \| \Psi_t(x_i) - \Psi_t(x_j) \|_2 \]
We can now use the diffusion maps \( \{\Psi_t(x_i)\}_i \) for classification instead of using the feature vectors themselves.
We can now use the diffusion maps \( \{\Psi_t(x_i)\}_i \) for classification instead of using the feature vectors themselves.

The Euclidean distances between the diffusion maps approximates the diffusion distances between the feature vectors.
We can now use the diffusion maps \( \{ \Psi_t(x_i) \}_i \) for classification instead of using the feature vectors themselves.

The Euclidean distances between the diffusion maps approximates the diffusion distances between the feature vectors

\[ \Rightarrow \text{Classification based on Euclidean distances is meaningful.} \]
We can now use the diffusion maps $\{\Psi_t(x_i)\}_i$ for classification instead of using the feature vectors themselves.

The Euclidean distances between the diffusion maps approximates the diffusion distances between the feature vectors

⇒ *Classification based on Euclidean distances is meaningful.*

Classification using k-nearest neighbors (k-NN) based on Euclidean distances, is improved when adding the stage of diffusion maps.
The experimental framework
The experimental framework

- Training set
- Feature extraction
- Diffusion mapping
The experimental framework

1. Training set
   - Feature extraction
   - Diffusion mapping

2. New sample
   - Embedding
   - Feature extraction
The experimental framework

- Training set
- Feature extraction
- Diffusion mapping
  - Diffusion mapping of the training set
  - K-nearest neighbors
- Embedding
- New sample
- Feature extraction
- Diffusion mapping of the new sample
The experimental framework

Training set → Feature extraction → Diffusion mapping → Embedding → New sample → Feature extraction → Diffusion mapping of the new sample → K-nearest neighbors → Classification of the new sample

Diffusion mapping of the training set → Classification of the new sample
Musical genre classification [Genussov & Cohen, 2010b]

- GTZAN genre data set
  (http://marsyas.info/download/data_sets) - includes 1000 songs, 100 from each genre (Jazz, Classic, Metal, Rock, etc.) - divided randomly into training set and testing set.
Musical genre classification [Genussov & Cohen, 2010b]

- GTZAN genre data set (http://marsyas.info/download/data_sets) - includes 1000 songs, 100 from each genre (Jazz, Classic, Metal, Rock, etc.) - divided randomly into training set and testing set.
- Mapping the feature vectors to diffusion mappings of 10-D.
Musical genre classification [Genussov & Cohen, 2010b]

- GTZAN genre data set
  (http://marsyas.info/download/data_sets) - includes 1000 songs, 100 from each genre (Jazz, Classic, Metal, Rock, etc.) - divided randomly into training set and testing set.
- Mapping the feature vectors to diffusion mappings of 10-D.
- For visualization, diffusion mapping of 2-D:

![Diffusion coordinates - 2D](image_url)
Musical genre classification [Genussov & Cohen, 2010b]

- GTZAN genre data set
  (http://marsyas.info/download/data_sets) - includes 1000 songs, 100 from each genre (Jazz, Classic, Metal, Rock, etc.) - divided randomly into training set and testing set.
- Mapping the feature vectors to diffusion mappings of 10-D.
- For visualization, diffusion mapping of 2-D:

![Diffusion coordinates - 2D](image)
### Musical genre classification - The confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>&quot;Blues&quot;</th>
<th>&quot;Classic&quot;</th>
<th>&quot;Country&quot;</th>
<th>&quot;Disco&quot;</th>
<th>&quot;Hiphop&quot;</th>
<th>&quot;Jazz&quot;</th>
<th>&quot;Metal&quot;</th>
<th>&quot;Pop&quot;</th>
<th>&quot;Reggae&quot;</th>
<th>&quot;Rock&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blues</td>
<td>0.60</td>
<td>0.00</td>
<td>0.09</td>
<td>0.04</td>
<td>0.05</td>
<td>0.02</td>
<td>0.07</td>
<td>0.00</td>
<td>0.04</td>
<td>0.10</td>
</tr>
<tr>
<td>Classic</td>
<td>0.01</td>
<td>0.81</td>
<td>0.05</td>
<td>0.01</td>
<td>0.00</td>
<td>0.08</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
</tr>
<tr>
<td>Country</td>
<td>0.07</td>
<td>0.02</td>
<td>0.48</td>
<td>0.05</td>
<td>0.01</td>
<td>0.11</td>
<td>0.00</td>
<td>0.06</td>
<td>0.07</td>
<td>0.13</td>
</tr>
<tr>
<td>Disco</td>
<td>0.02</td>
<td>0.00</td>
<td>0.08</td>
<td>0.37</td>
<td>0.11</td>
<td>0.03</td>
<td>0.07</td>
<td>0.12</td>
<td>0.06</td>
<td>0.14</td>
</tr>
<tr>
<td>Hiphop</td>
<td>0.04</td>
<td>0.00</td>
<td>0.02</td>
<td>0.08</td>
<td>0.52</td>
<td>0.00</td>
<td>0.02</td>
<td>0.13</td>
<td>0.16</td>
<td>0.02</td>
</tr>
<tr>
<td>Jazz</td>
<td>0.06</td>
<td>0.12</td>
<td>0.10</td>
<td>0.04</td>
<td>0.00</td>
<td>0.54</td>
<td>0.01</td>
<td>0.04</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>Metal</td>
<td>0.06</td>
<td>0.00</td>
<td>0.01</td>
<td>0.08</td>
<td>0.02</td>
<td>0.00</td>
<td>0.73</td>
<td>0.00</td>
<td>0.01</td>
<td>0.10</td>
</tr>
<tr>
<td>Pop</td>
<td>0.00</td>
<td>0.00</td>
<td>0.07</td>
<td>0.08</td>
<td>0.06</td>
<td>0.02</td>
<td>0.00</td>
<td>0.68</td>
<td>0.08</td>
<td>0.03</td>
</tr>
<tr>
<td>Reggae</td>
<td>0.05</td>
<td>0.00</td>
<td>0.07</td>
<td>0.05</td>
<td>0.07</td>
<td>0.01</td>
<td>0.01</td>
<td>0.07</td>
<td>0.62</td>
<td>0.05</td>
</tr>
<tr>
<td>Rock</td>
<td>0.06</td>
<td>0.00</td>
<td>0.19</td>
<td>0.09</td>
<td>0.04</td>
<td>0.10</td>
<td>0.11</td>
<td>0.05</td>
<td>0.05</td>
<td>0.30</td>
</tr>
</tbody>
</table>
Musical genre classification - comparison to other dimensionality reduction methods

<table>
<thead>
<tr>
<th>Number of classes</th>
<th>No dimensionality reduction</th>
<th>PCA</th>
<th>Diffusion Maps</th>
<th>Laplacian Eigenmaps</th>
<th>LLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>61.9%</td>
<td>60.3%</td>
<td><strong>56.3%</strong></td>
<td>51.7%</td>
<td>55.8%</td>
</tr>
<tr>
<td>5</td>
<td>87.5%</td>
<td>86.7 ± 4.5%</td>
<td><strong>83.4%</strong></td>
<td>83.2%</td>
<td>85.5%</td>
</tr>
<tr>
<td>2</td>
<td>98.4%</td>
<td>97.8%</td>
<td><strong>97.5%</strong></td>
<td>91.0%</td>
<td>96.3%</td>
</tr>
<tr>
<td>5 groups of pairs</td>
<td>69.1%</td>
<td>67.4%</td>
<td><strong>65.7%</strong></td>
<td>60.3%</td>
<td>65.7%</td>
</tr>
</tbody>
</table>
Unvoiced fricative phonemes identification [Genussov, Lavner & Cohen, 2010]

- The phonemes /s/, /sh/, /f/, /th/ (254 for each kind), from TIMIT data base - divided randomly into training set and testing set.
Unvoiced fricative phonemes identification [Genussov, Lavner & Cohen, 2010]

- The phonemes /s/, /sh/, /f/, /th/ (254 for each kind), from TIMIT data base - divided randomly into training set and testing set.

- Mapping the feature vectors to diffusion mappings of 10-D.
The phonemes /s/, /sh/, /f/, /th/ (254 for each kind), from TIMIT data base - divided randomly into training set and testing set.

Mapping the feature vectors to diffusion mappings of 10-D.

For visualization, diffusion mapping to 2-D:
Unvoiced fricative phonemes identification - the confusion matrix

**Table:** Classification of phonemes - majority based

<table>
<thead>
<tr>
<th></th>
<th>&quot;[f]&quot;</th>
<th>&quot;[s]&quot;</th>
<th>&quot;[sh]&quot;</th>
<th>&quot;[th]&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>[f]</td>
<td>0.87</td>
<td>0.01</td>
<td>0.03</td>
<td>0.09</td>
</tr>
<tr>
<td>[s]</td>
<td>0.02</td>
<td>0.87</td>
<td>0.11</td>
<td>0.00</td>
</tr>
<tr>
<td>[sh]</td>
<td>0.06</td>
<td>0.09</td>
<td>0.85</td>
<td>0.00</td>
</tr>
<tr>
<td>[th]</td>
<td>0.27</td>
<td>0.06</td>
<td>0.01</td>
<td>0.66</td>
</tr>
</tbody>
</table>

**Table:** Classification of segments

<table>
<thead>
<tr>
<th></th>
<th>&quot;[f]&quot;</th>
<th>&quot;[s]&quot;</th>
<th>&quot;[sh]&quot;</th>
<th>&quot;[th]&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>[f]</td>
<td>0.70</td>
<td>0.04</td>
<td>0.06</td>
<td>0.20</td>
</tr>
<tr>
<td>[s]</td>
<td>0.07</td>
<td>0.68</td>
<td>0.16</td>
<td>0.09</td>
</tr>
<tr>
<td>[sh]</td>
<td>0.10</td>
<td>0.09</td>
<td>0.79</td>
<td>0.02</td>
</tr>
<tr>
<td>[th]</td>
<td>0.34</td>
<td>0.15</td>
<td>0.04</td>
<td>0.47</td>
</tr>
</tbody>
</table>
Unvoiced fricative phonemes identification - Comparison between dimensionality reduction methods

<table>
<thead>
<tr>
<th>Dimensionality reduction type</th>
<th>No dimensionality reduction</th>
<th>Diffusion maps</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification of segments</td>
<td>68.8%</td>
<td>66.6%</td>
<td>68.6%</td>
</tr>
<tr>
<td>Classification of phonemes - majority-based decision</td>
<td>81.0%</td>
<td>78.6%</td>
<td>81.0%</td>
</tr>
</tbody>
</table>
Outline

1. Background and motivation
2. Transcription of polyphonic music by Sparse Representations
3. Classification of audio and speech data using geometric methods
4. Conclusions
   - Summary
   - Future research
Transcription of polyphonic music

- Developing the Musically-Structured (MS) dictionary learning algorithm for transcription of polyphonic music.
  - An unsupervised method.
  - Outperforms explicit dictionaries
    - More computationally efficient.
    - Avoids over-fitting and recognizes chords better.
  - Represents the signal better than analytic dictionaries.
  - This algorithm outperforms regular methods of sparse representations especially in the case of a small data set with multiple overlapping harmonics.
Classification of audio and speech data

- We assume that feature vectors of audio and speech data lie on a non-linear, low-dimensional manifold.
- Integration of a non-linear manifold learning technique namely “Diffusion maps” into traditional classification methods.
- Using Euclidean distances for classification after the diffusion mapping is theoretically meaningful.
- No significant improvement in classification results in our cases, perhaps due to the selection of features, for these specific applications.
Next research

- Transcription of polyphonic music
  - Build the structured dictionary based on real piano music ("stretched octaves")
  - Add more atoms to represent each note in the dictionary
  - Use high-level prior information
  - Transcription of several instruments
Future research

- Transcription of polyphonic music
  - Build the structured dictionary based on real piano music ("stretched octaves")
  - Add more atoms to represent each note in the dictionary
  - Use high-level prior information
  - Transcription of several instruments

- Classification of audio data
  - Extract more features
  - Other applications of audio and speech classification
  - Examine the robustness of the algorithm to noise