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

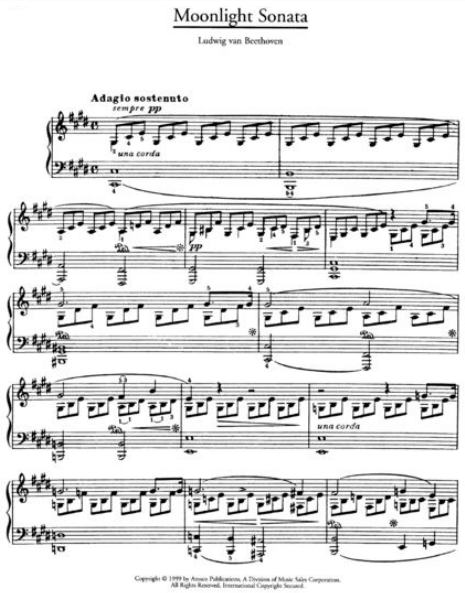
Polyphonic music

- More than one note is played at a time
- Can be produced from a single instrument or from several instruments

Transcription of music

Definition: Transcription of music

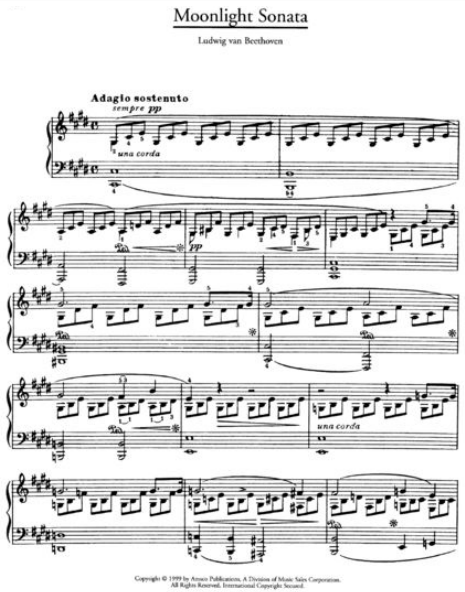
Identification of musical parameters in an audio signal which are required to write down the score sheet of the notes



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

Classification of audio and speech data

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?

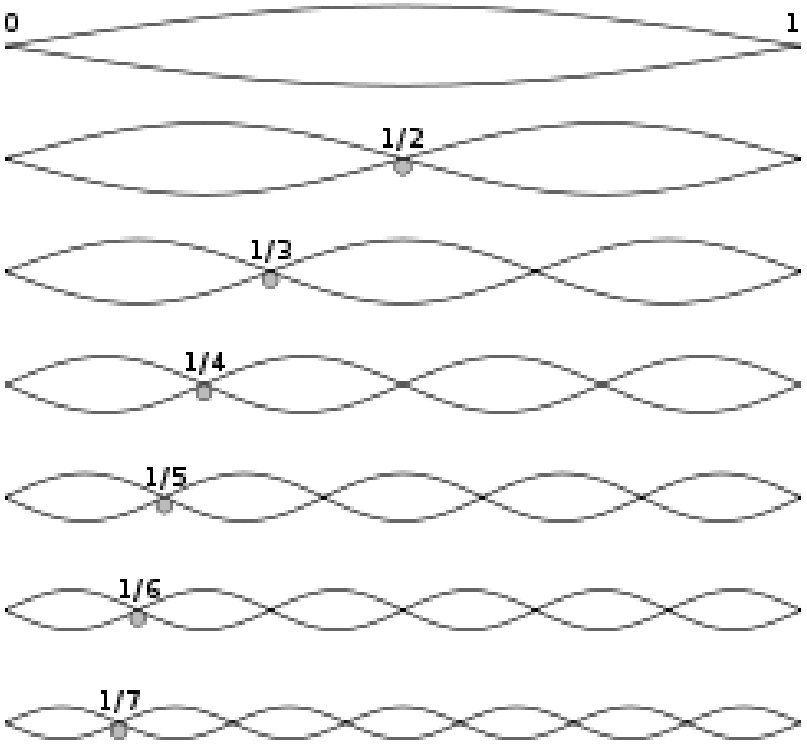
- Structure and organize large data bases (on the web, for radio stations and more)
- Similarity retrieval
- Audio thumbnailing
- Smart video conference systems
- Advanced hearing aids

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

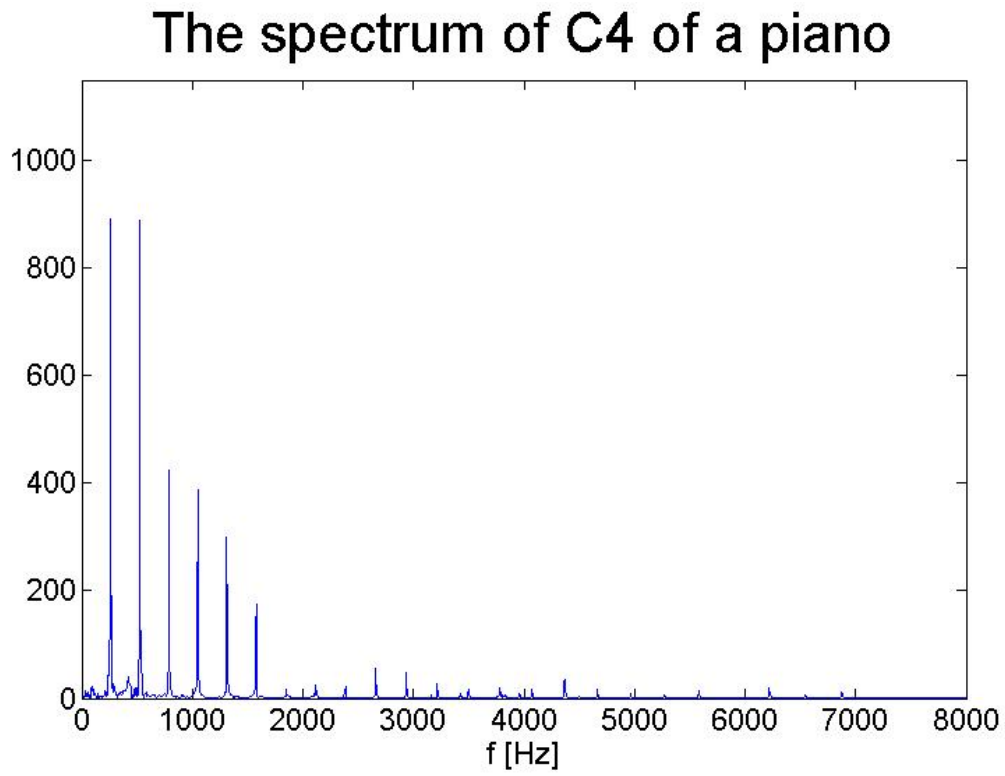
Harmonics

Harmonic instruments act as periodic oscillators, leading to vibrations in a fundamental frequency (pitch) and in its integer multiples, i.e. its *harmonics*.



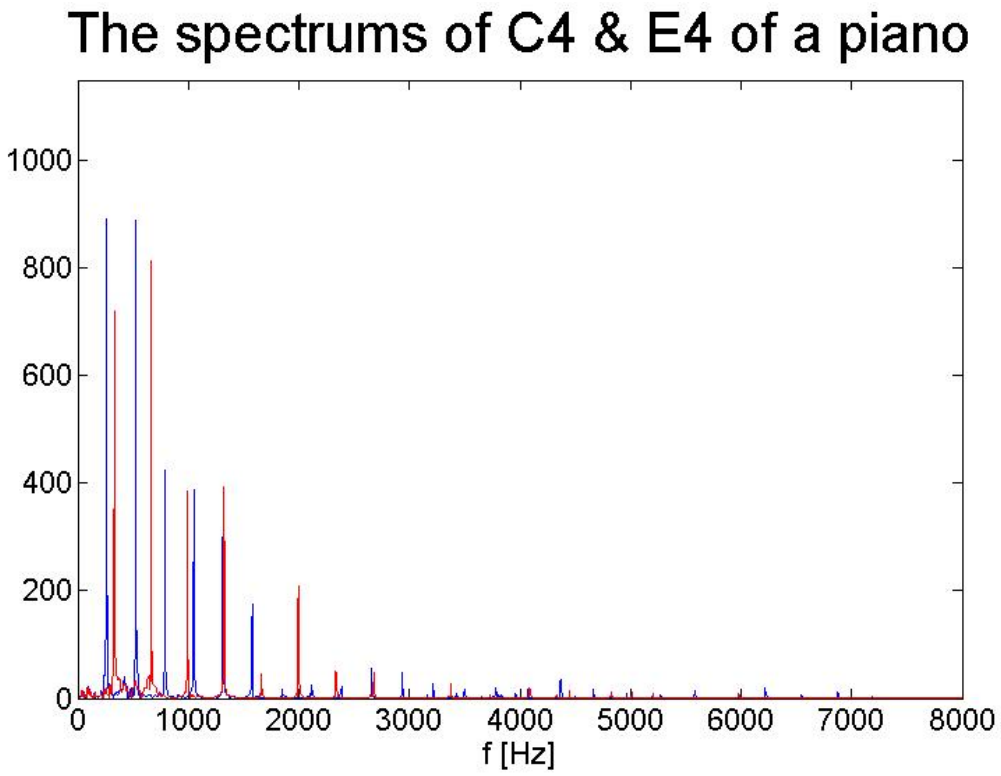
The difficulty in transcription of polyphonic music

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



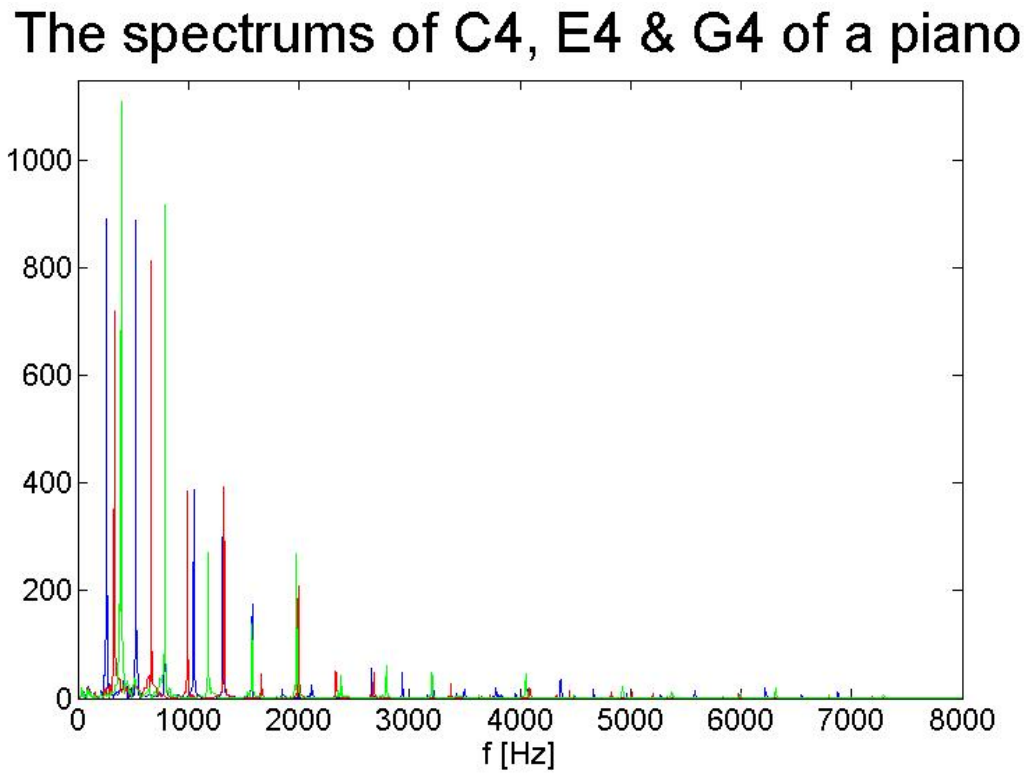
The difficulty in transcription of polyphonic music

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



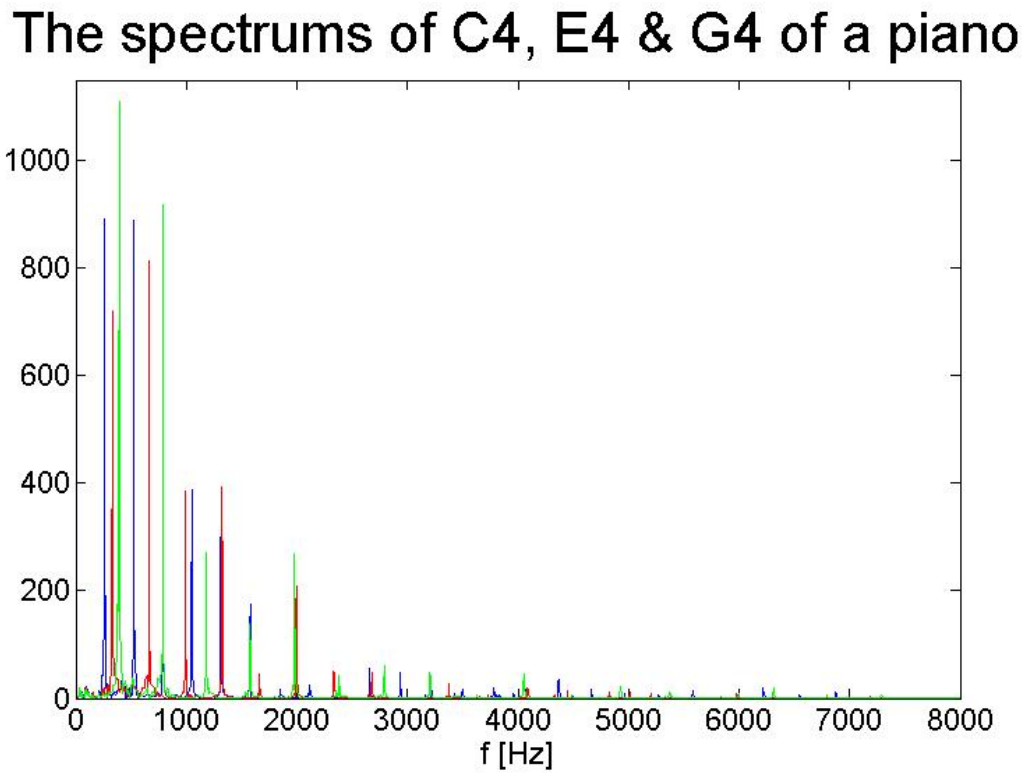
The difficulty in transcription of polyphonic music

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



The difficulty in transcription of polyphonic music

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



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

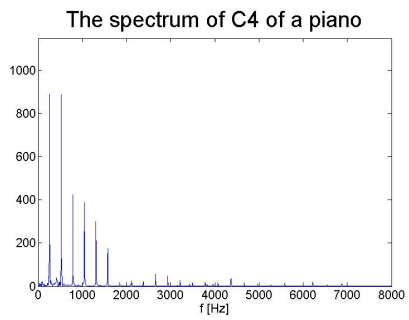
- Require a diverse and large data set

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

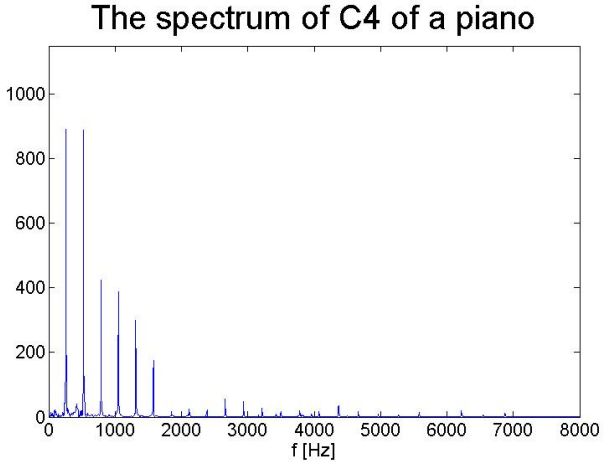
- Require a diverse and large data set
- The assumption of a common distribution of the intensities of the harmonics is inaccurate
  $\Rightarrow$ The intensities of the harmonics change unevenly according to
  - The musical instrument
  - The intensity
  - Time
  - The fundamental frequency



Our solution

We offer an algorithm which

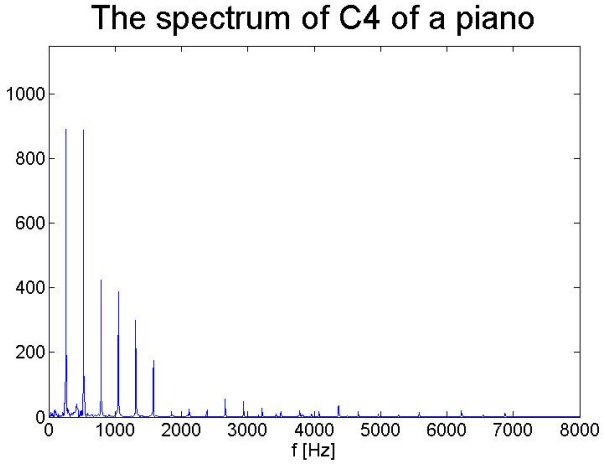
- Learns the intensities of the harmonics from the signal itself.



Our solution

We offer an algorithm which

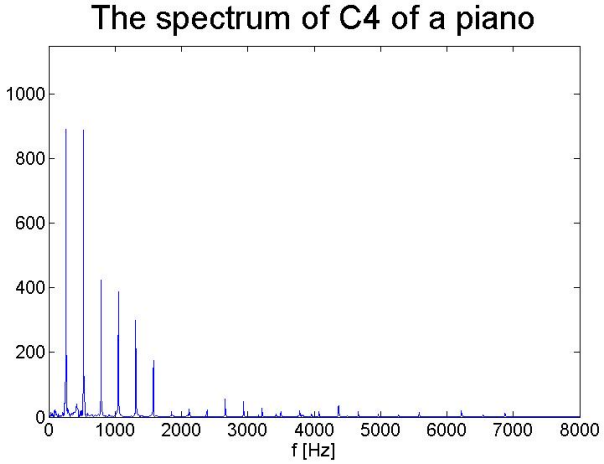
- Learns the intensities of the harmonics from the signal itself.
- Is based on the common structure of music spectra, thus does not require a diverse or large data set.



Our solution

We offer an algorithm which

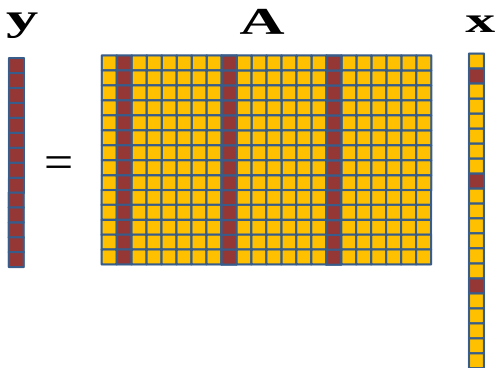
- Learns the intensities of the harmonics from the signal itself.
- Is based on the common structure of music spectra, thus does not require a diverse or large data set.
- Is based on sparse representations with a suitable parametric dictionary.



Sparse coding

Sparse coding is formulated as solving the problem

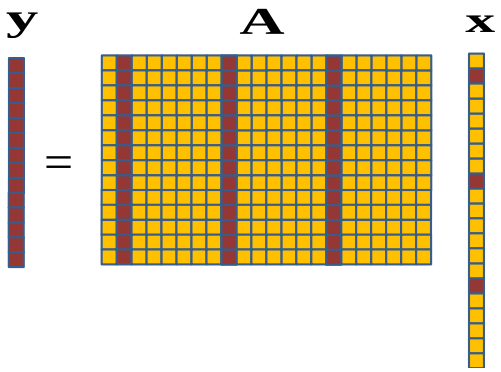
$$P_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{y} \tag{1}$$



Sparse coding

Sparse coding is formulated as solving the problem

$$P_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{y} \tag{1}$$



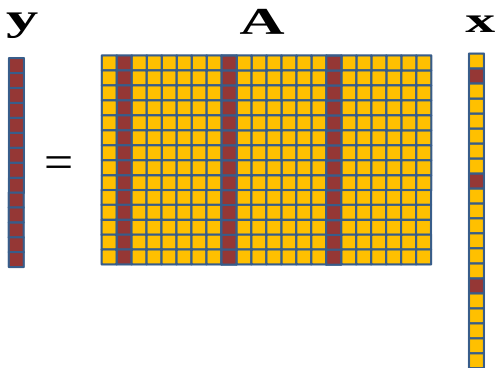
In an approximated formulation

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \leq K \tag{2}$$

Sparse coding

Sparse coding is formulated as solving the problem

$$P_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{y} \tag{1}$$



In an approximated formulation

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \leq K \tag{2}$$

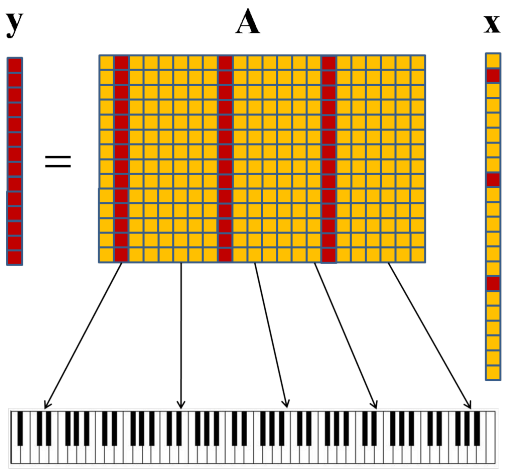
- The sparse coding problem is NP-hard
  $\Rightarrow$ Approximation using a pursuit algorithm.

Sparse coding of music

- The number of notes played at a time is limited (K)

Sparse coding of music

- 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.



Pursuit algorithms for solving the sparse coding

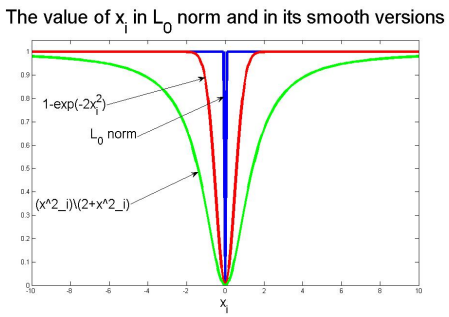
Several types of pursuit algorithms:

- Greedy methods - Least-Squares Orthogonal Matching Pursuit (LS-OMP), Orthogonal Matching Pursuit (OMP), Matching Pursuit (MP), Weak-MP and Thresholding.

Pursuit algorithms for solving the sparse coding

Several types of pursuit algorithms:

- Greedy methods - Least-Squares Orthogonal Matching Pursuit (LS-OMP), Orthogonal Matching Pursuit (OMP), Matching Pursuit (MP), Weak-MP and Thresholding.
- Relaxation methods - Replace the ℓ_0 "norm" by
  - Other ℓ_p norms for some $p \in (0, 1]$ - FOCUSS, Basis Pursuit
  - Smooth functions such as $\sum_{i=1}^{m} 1 - e^{-\alpha x_i^2}$ or $\sum_{i=1}^{m} x_i^2/(\alpha + x_i^2)$



Choosing a dictionary

There are several possibilities for choosing a dictionary **A**:

1. Analytic dictionaries
2. Learned dictionaries
3. Parametric dictionaries

Analytic 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

Analytic dictionaries

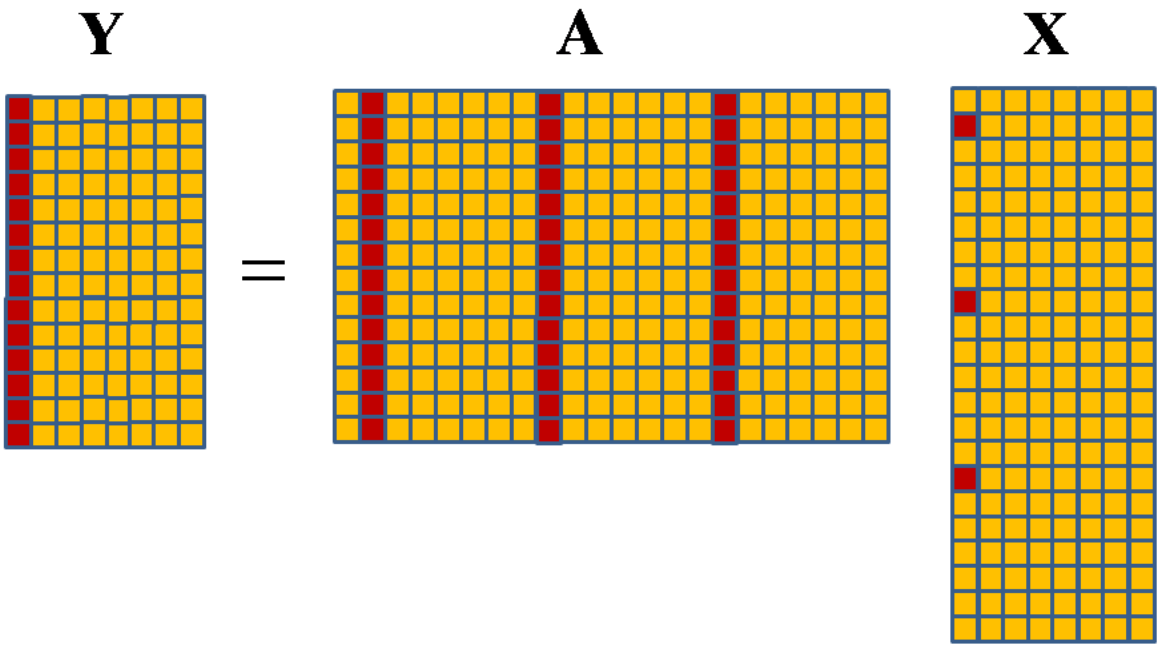
- The dictionary **A** is pre-defined and constant.
- Examples for transforms under which images were found to be sparse:
  - Curvelets [Candes, 2000]
  - Contourlets [Do & Vetterli, 2005]
  - Wavelets [Mallat, 1993]
  - Short-time Fourier transform

$\Rightarrow$ 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

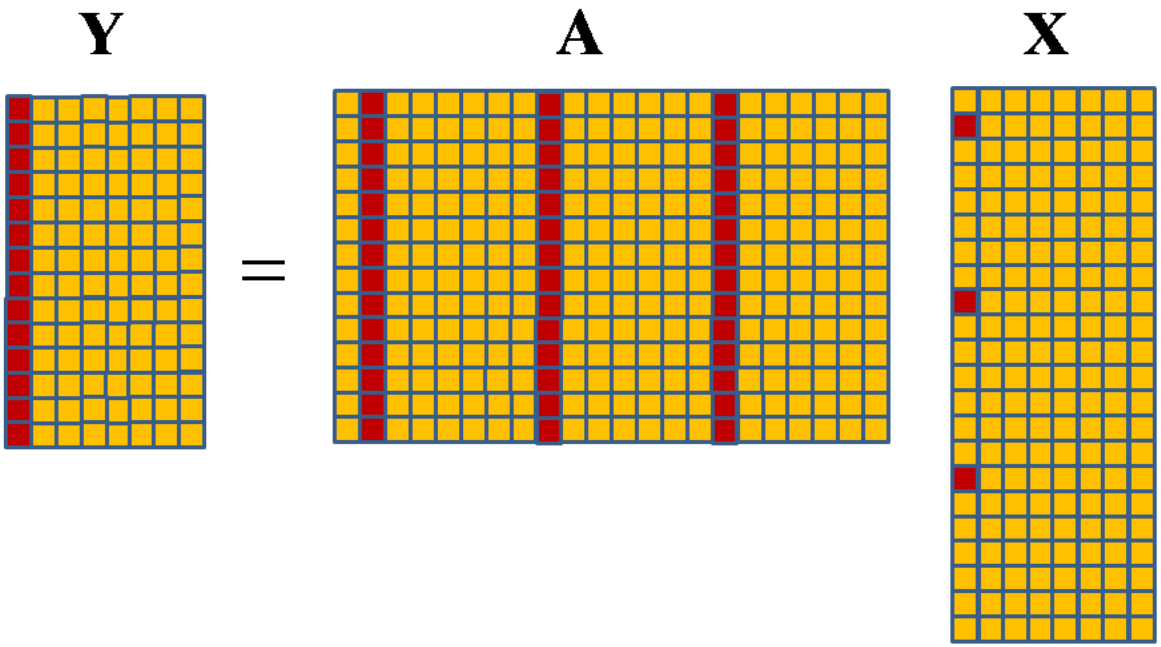
$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \leq K$$



Learned dictionaries [Olshausen & Field, 1996]

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

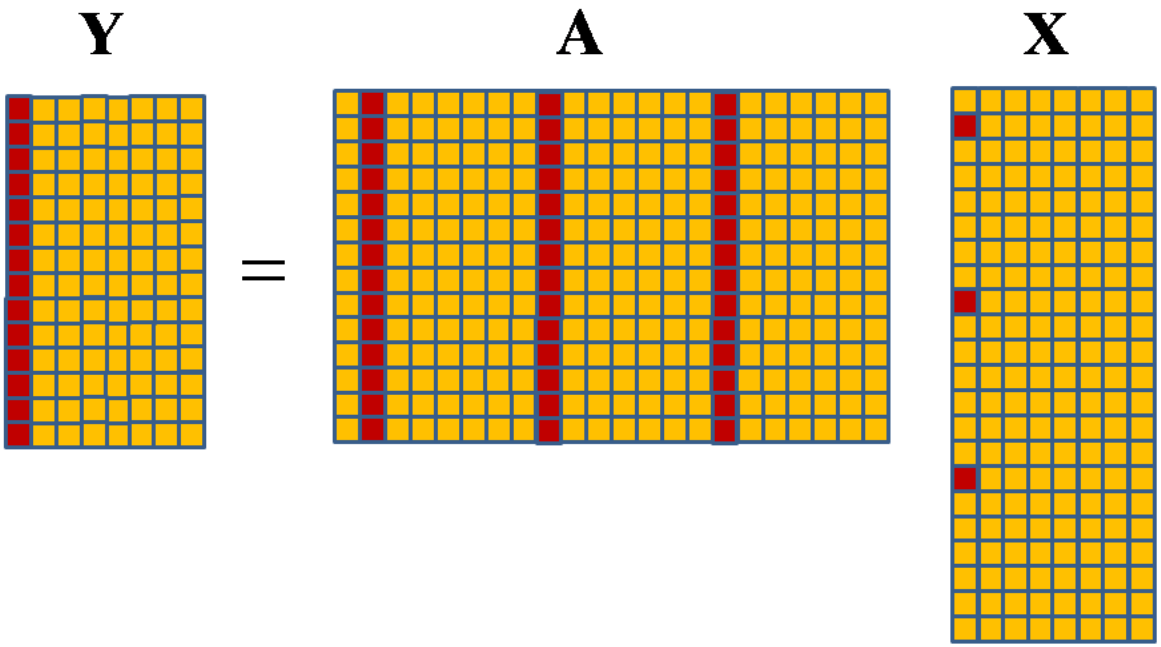
$$\min_{\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

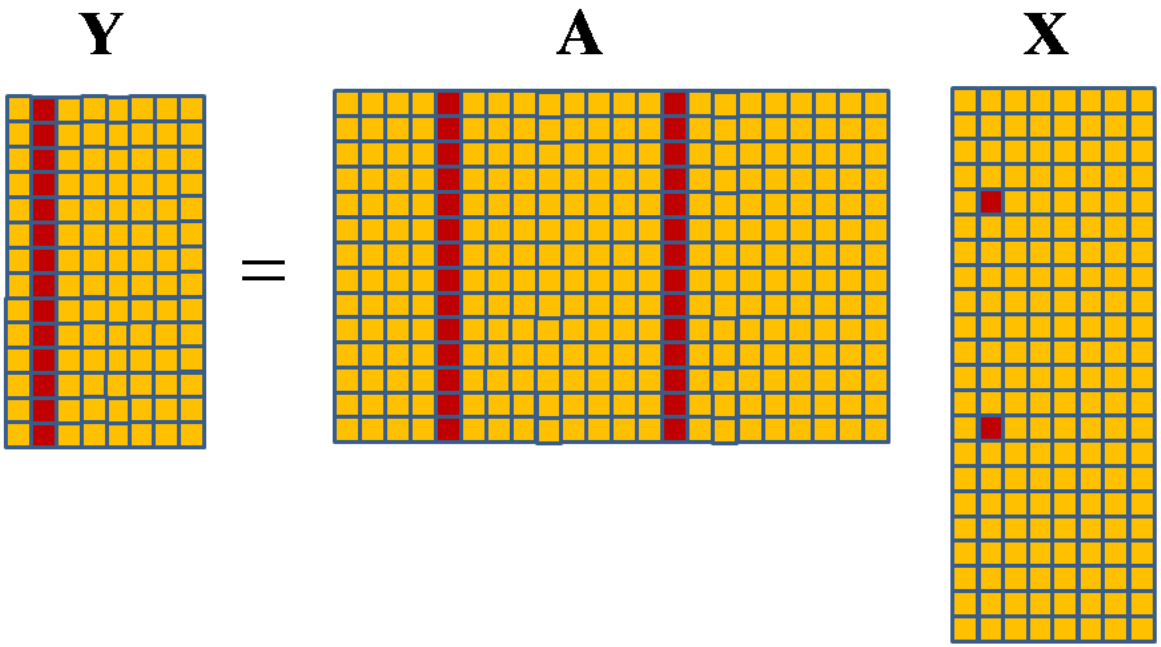
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

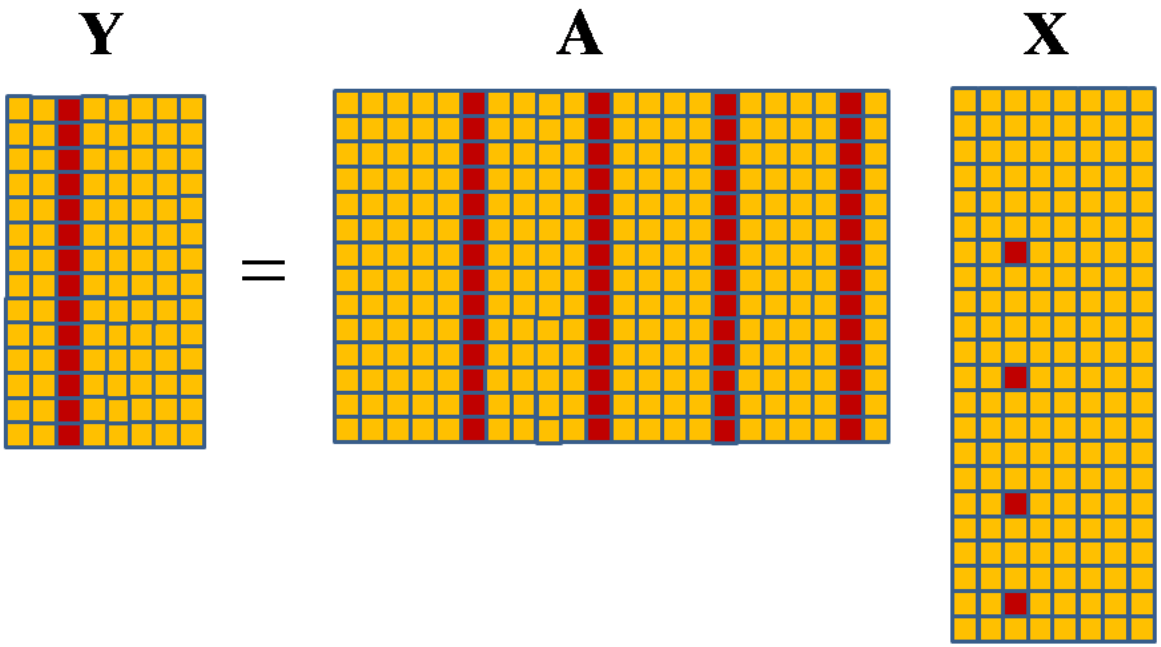
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

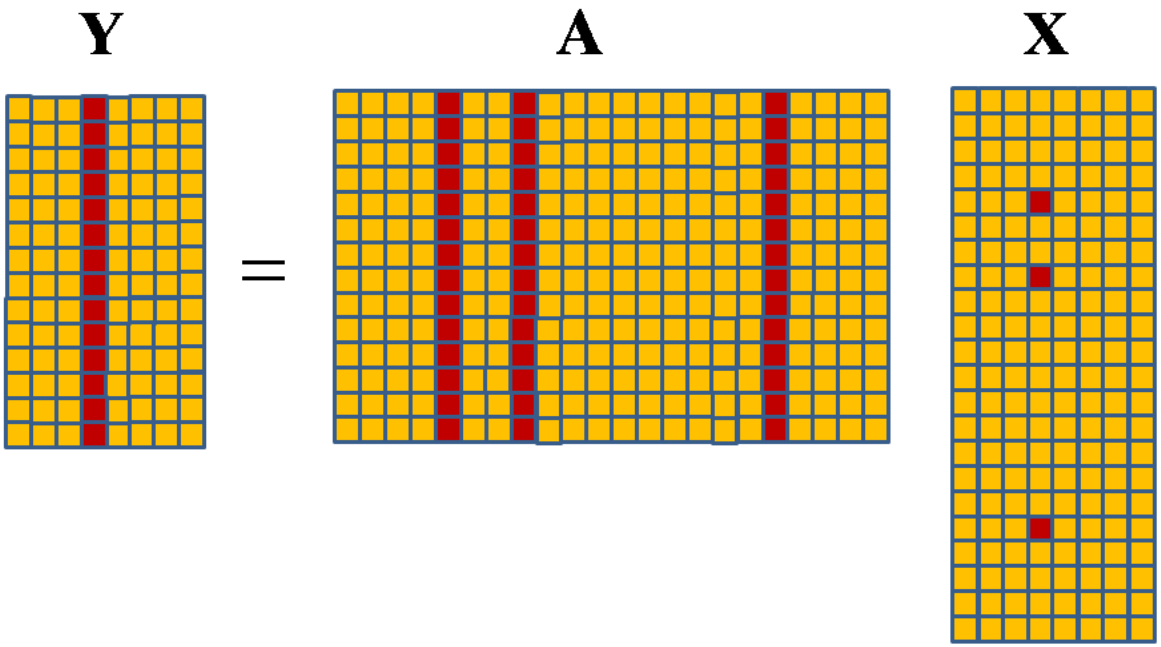
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

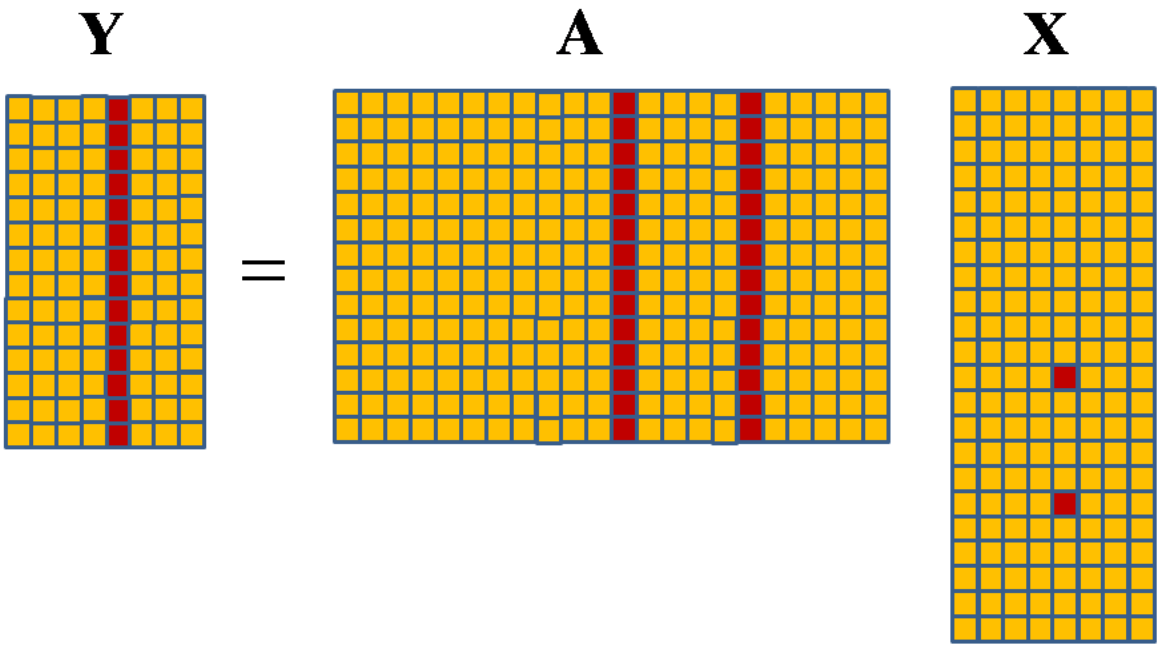
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

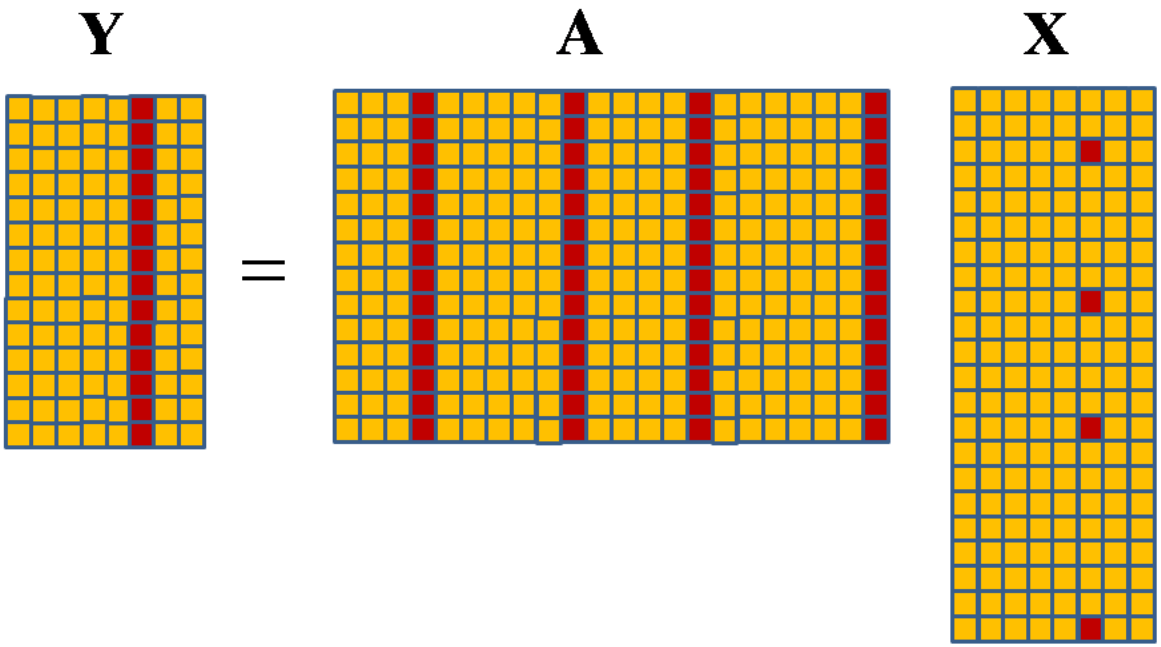
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

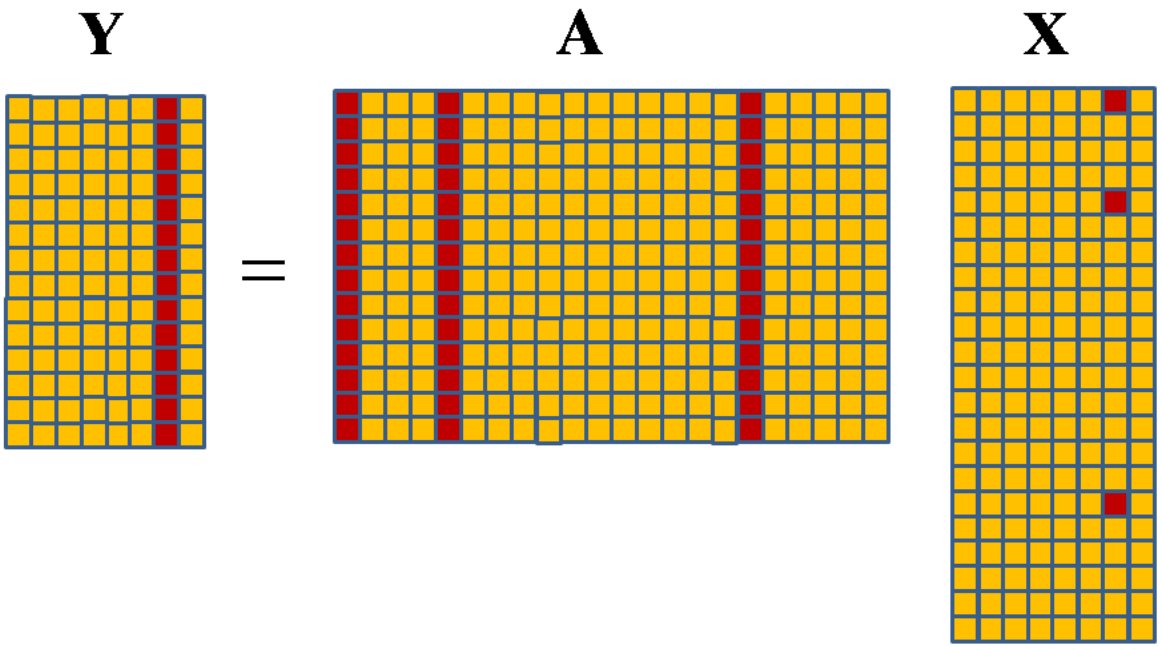
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

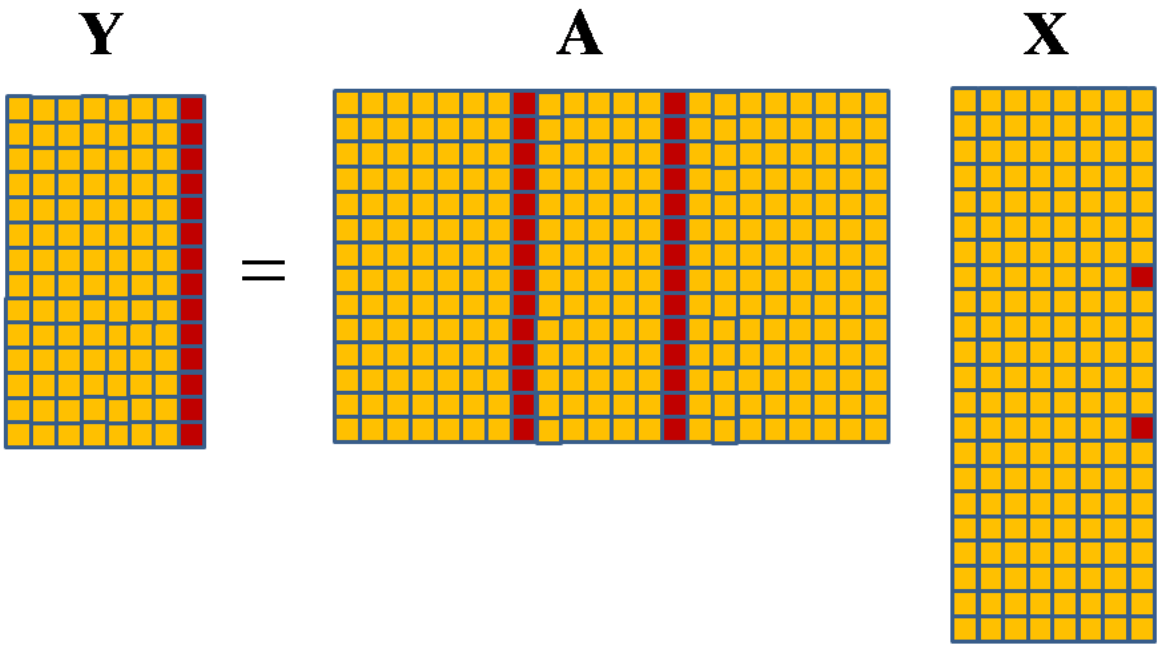
$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{AX} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K_i$$



Learned dictionaries [Olshausen & Field, 1996]

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

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



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:

$$\mathbf{A}_{(\mathbf{k})} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}_{(\mathbf{k})}\|_{\mathbf{F}}^{\mathbf{2}} = \mathbf{Y}\mathbf{X}^{\dagger}{}_{(\mathbf{k})}$$

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:

$$\mathbf{A}_{(\mathbf{k})} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}_{(\mathbf{k})}\|_{\mathbf{F}}^{\mathbf{2}} = \mathbf{Y}\mathbf{X}^{\dagger}{}_{(\mathbf{k})}$$

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:

$$\mathbf{A}_{(\mathbf{k})} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}_{(\mathbf{k})}\|_{\mathbf{F}}^{\mathbf{2}} = \mathbf{Y}\mathbf{X}^{\dagger}{}_{(\mathbf{k})}$$

Stopping criterion : if the change in $\|\mathbf{Y} - \mathbf{A}_{(\mathbf{k})}\mathbf{X}_{(\mathbf{k})}\|_F^2$ is small enough, stop.

K-SVD [Aharon et al., 2006]

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 $\|\mathrm{Y} - \mathrm{A}_{(k)}\mathrm{X}_{(k)}\|_F^2$ 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 $\|\mathrm{Y} - \mathrm{A}_{(k)}\mathrm{X}_{(k)}\|_F^2$ is small enough, stop.

K-SVD [Aharon et al., 2006]

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

K-SVD [Aharon et al., 2006]

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

Define the error matrix

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

K-SVD [Aharon et al., 2006]

K-SVD **Dictionary update**
For the 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 $\mathbf{E}_{j_0}$ only to the columns corresponding to the signals in $\mathbf{Y}$ that use $\mathbf{a}_{j_0}$ and get $\mathbf{E}_{j_0}^R$

K-SVD [Aharon et al., 2006]

K-SVD **Dictionary update**
For the 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 $\mathbf{E}_{j_0}$ only to the columns corresponding to the signals in $\mathbf{Y}$ that use $\mathbf{a}_{j_0}$ and get $\mathbf{E}_{j_0}^R$

Apply SVD decomposition of the matrix $\mathbf{E}_{j_0}^R$

K-SVD [Aharon et al., 2006]

K-SVD **Dictionary update**
For the 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 $\mathbf{E}_{j_0}$ only to the columns corresponding to the signals in $\mathbf{Y}$ that use $\mathbf{a}_{j_0}$ and get $\mathbf{E}_{j_0}^R$

Apply SVD decomposition of the matrix $\mathbf{E}_{j_0}^R$

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

Parametric dictionaries

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

Parametric dictionaries

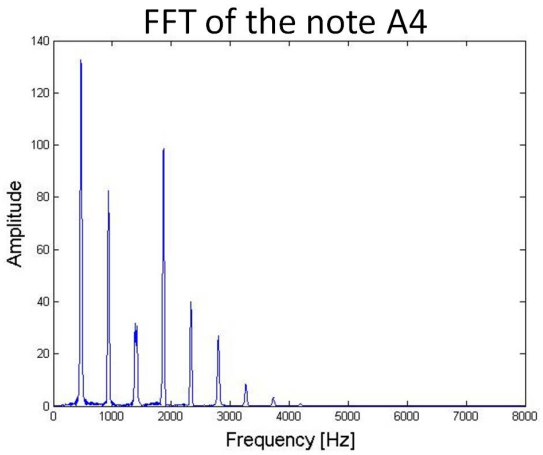
- A compromise between analytic and learned dictionaries
- Structured dictionaries, in which a limited set of parameters are learned

Examples:

- The double sparsity model [Rubinstein, Zibulevsky & Elad, 2010]
- The image signature dictionary [Aharon & Elad, 2008]

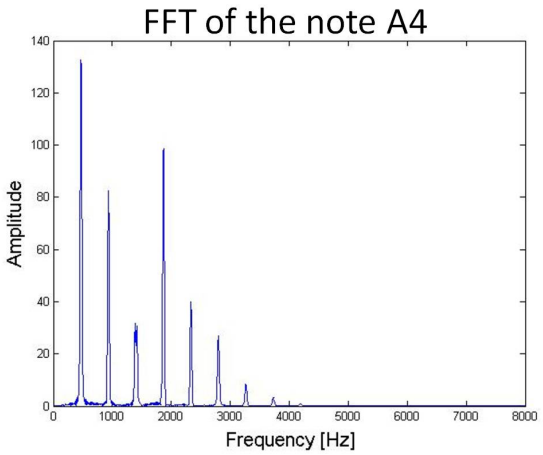
A parametric dictionary for music spectra

- Musical tones share a common structure of their spectra:



A parametric dictionary for music spectra

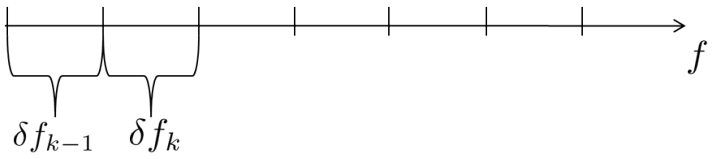
- Musical tones share a common structure of their spectra:



- 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").

The Constant Q transform (CQT)

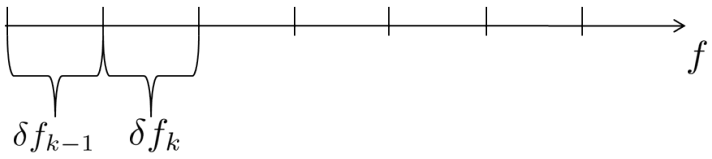
- The Short-time Fourier transform (STFT) - a series of linearly spaced filters.



$$\delta f_k = const$$

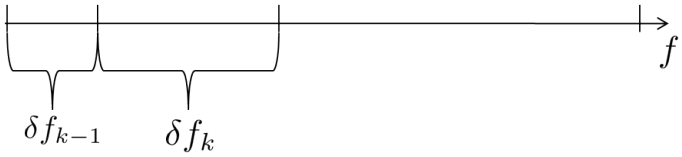
The Constant Q transform (CQT)

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



$$\delta f_k = const$$

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



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

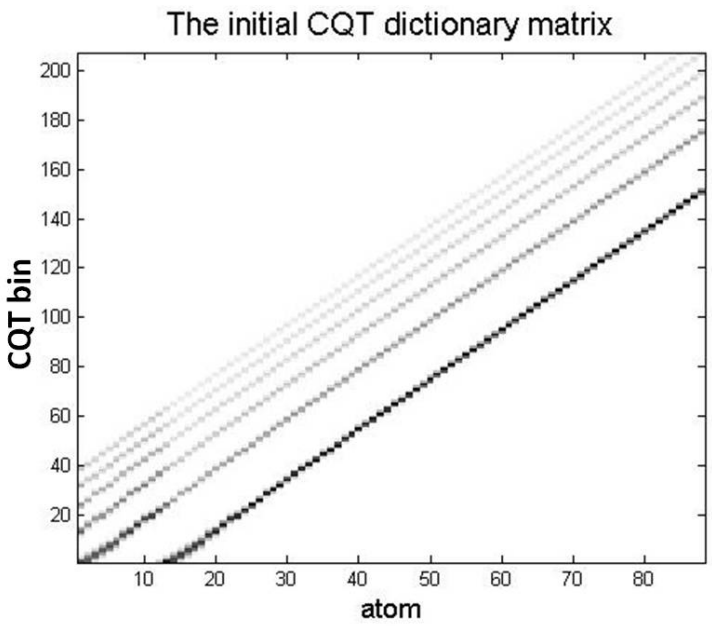
  - $n =$ the number of filters per octave
  - $Q = \frac{f_k}{\delta f_k}$

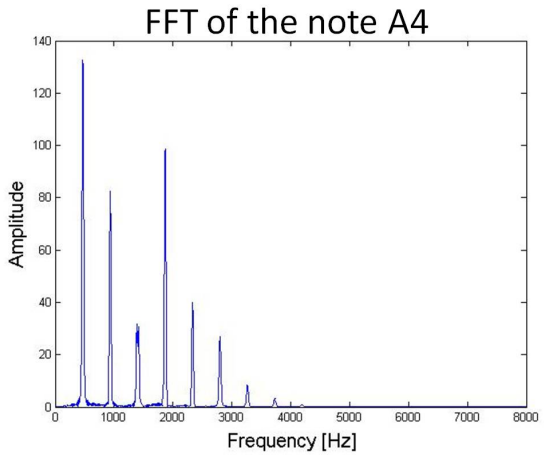
The Musically-Structured (MS) dictionary [Genussov & Cohen, 2010a]

- 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]

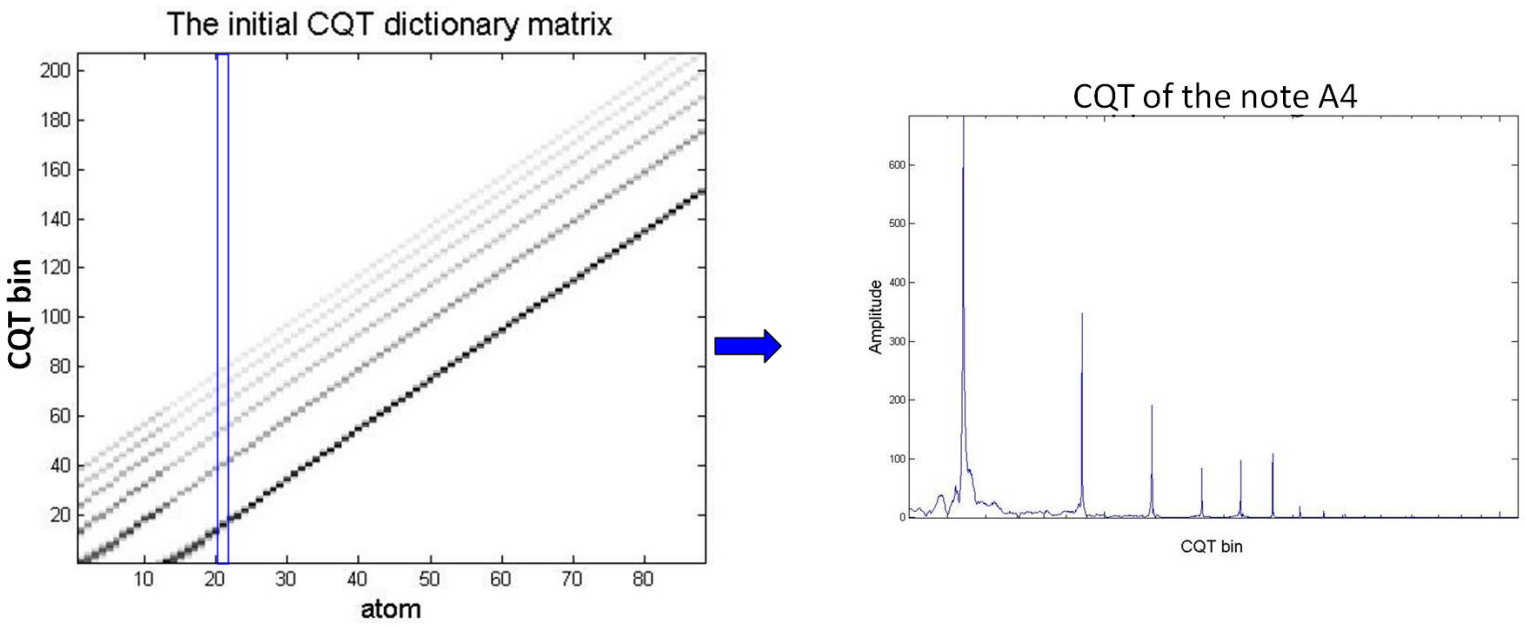
- We define a dictionary where each atom corresponds to the CQT of a different note in the piano (total 88 atoms).
  - The location of the support (non zero elements) is constant.
  - The entries in the support are learned.





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
  $\Rightarrow$ Doesn't require a large and diverse data set
- The values of the entries in the support are learned
  $\Rightarrow$ Adjustment to the *timbre* of the signal

Advantages of the MS dictionary

The complexity of the dictionary is proportional to its learned parameters

	Explicit dictionary	Musically structured dictionary
Complexity	$n \cdot m$	$h \cdot m$

- 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 $\|\mathbf{Y} - \mathbf{A}_{(\mathbf{k})}\mathbf{X}_{(\mathbf{k})}\|_F^2$ 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 $\|\mathbf{Y} - \mathbf{A}_{(\mathbf{k})}\mathbf{X}_{(\mathbf{k})}\|_F^2$ is small enough, stop.

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

MS-MOD Dictionary update:

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

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

MS-MOD Dictionary update:

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

Least squares:

$$\tilde{\mathbf{A}}_{(k)} = \underset{\tilde{\mathbf{A}}}{\operatorname{argmin}} \|\mathbf{Y} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}_{(\mathbf{k})}\|_F^2 = \mathbf{Y}\tilde{\mathbf{X}}_{(k)}^{\dagger}$$

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

MS-MOD Dictionary update:

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

Least squares:

$$\tilde{\mathbf{A}}_{(k)} = \underset{\tilde{\mathbf{A}}}{\operatorname{argmin}} \|\mathbf{Y} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}_{(\mathbf{k})}\|_F^2 = \mathbf{Y}\tilde{\mathbf{X}}_{(k)}^{\dagger}$$

Zero the entries out of the support of $\tilde{\mathbf{A}}$

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

MS-MOD Dictionary update:

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

Least squares:

$$\tilde{\mathbf{A}}_{(k)} = \underset{\tilde{\mathbf{A}}}{\operatorname{argmin}} \|\mathbf{Y} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}_{(\mathbf{k})}\|_F^2 = \mathbf{Y}\tilde{\mathbf{X}}_{(k)}^{\dagger}$$

Zero the entries out of the support of $\tilde{\mathbf{A}}$

Add the unused atoms to get the updated $\mathbf{A}$

Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

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

Sparse coding by a pursuit algorithm

MS-K-SVD Dictionary update

Stopping criterion : if the change in $\|\mathbf{Y} - \mathbf{A}_{(\mathbf{k})}\mathbf{X}_{(\mathbf{k})}\|_F^2$ is small enough, stop.

Musically-Structured K-SVD (MS-K-SVD) [Genussov & Cohen, 2010a]

MS-K-SVD Dictionary update

For the used atom $\mathbf{a}_{j_0}$:

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

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 $\mathbf{E}_{j_0}^R$, and *restrict the rows according to the support* to get $\tilde{\mathbf{E}}_{j_0}^R$

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 $\mathbf{E}_{j_0}^R$, and *restrict the rows according to the support* to get $\tilde{\mathbf{E}}_{j_0}^R$

→ Apply SVD decomposition of the matrix $\tilde{\mathbf{E}}_{j_0}^R$

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 $\mathbf{E}_{j_0}^R$, and *restrict the rows according to the support* to get $\tilde{\mathbf{E}}_{j_0}^R$

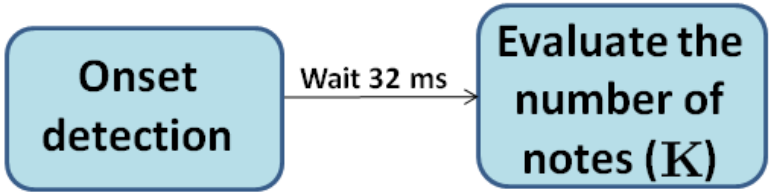
Apply SVD decomposition of the matrix $\tilde{\mathbf{E}}_{j_0}^R$

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

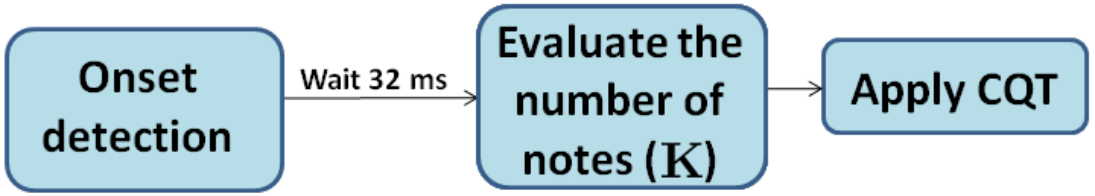
The musically-structured algorithm for transcription

The musically-structured algorithm for transcription

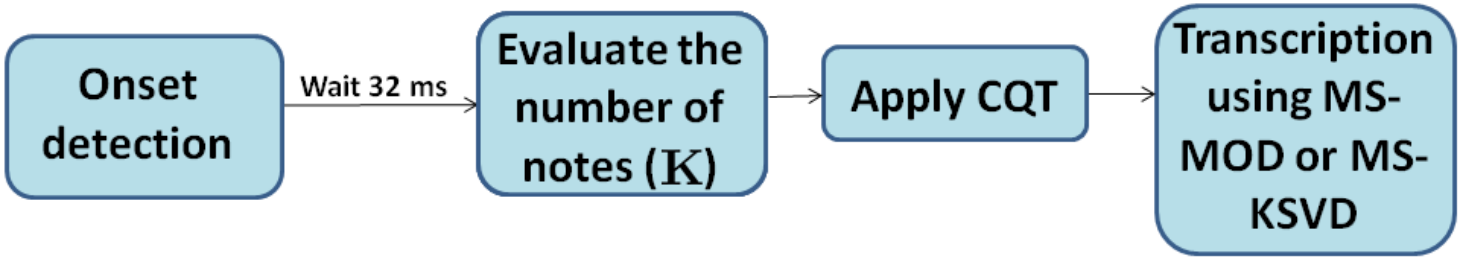
The musically-structured algorithm for transcription



The musically-structured algorithm for transcription

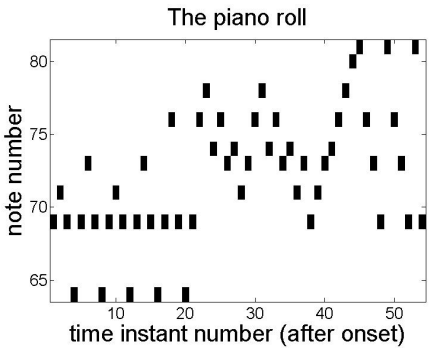


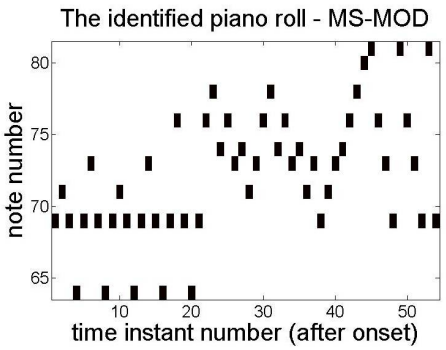
The musically-structured algorithm for transcription



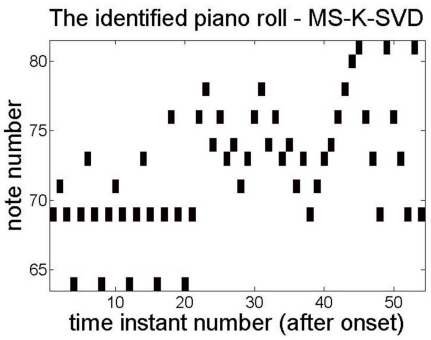
Synthesized MIDI music

- Monophonic music





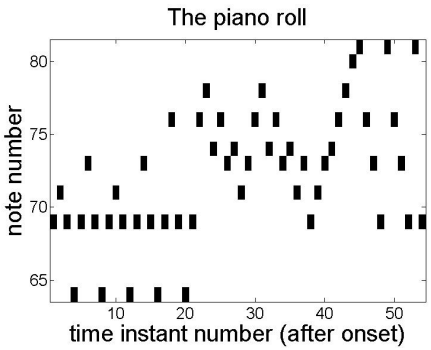
Accuracy=100%

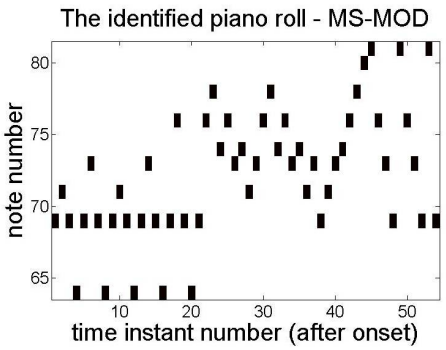


Accuracy=100%

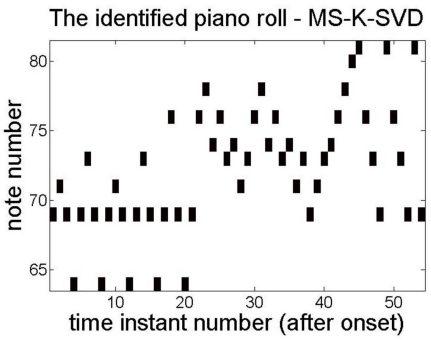
Synthesized MIDI music

- Monophonic music





Accuracy=100%



Accuracy=100%

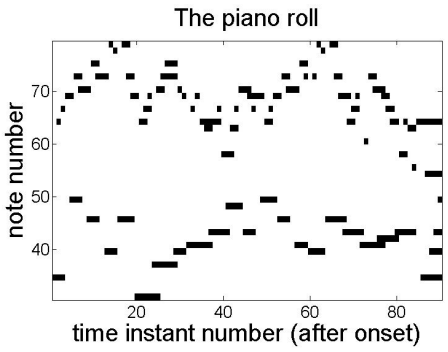
Definition [Dixon, 2000]

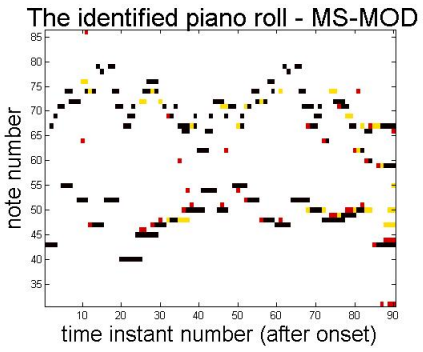
$$Accuracy = \frac{TP}{TP+FP+FN}$$

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

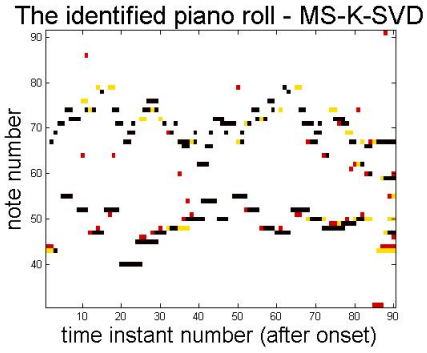
Synthesized MIDI music

- Simple polyphonic music





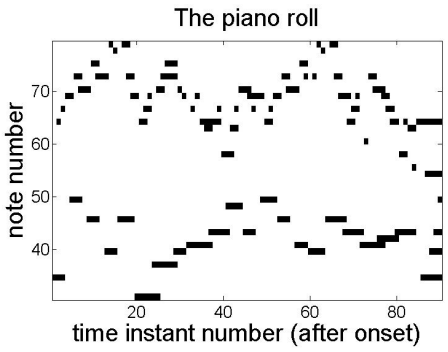
Accuracy=69.6%

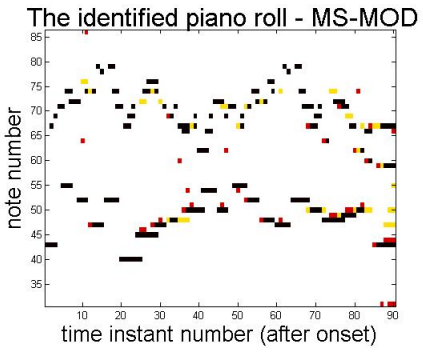


Accuracy=67.7%

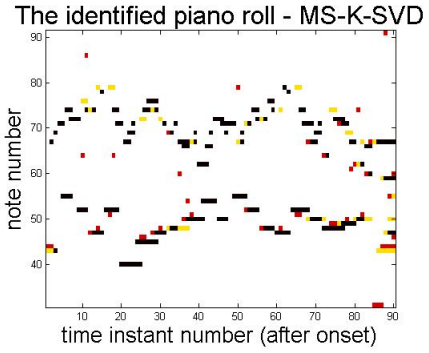
Synthesized MIDI music

- Simple polyphonic music



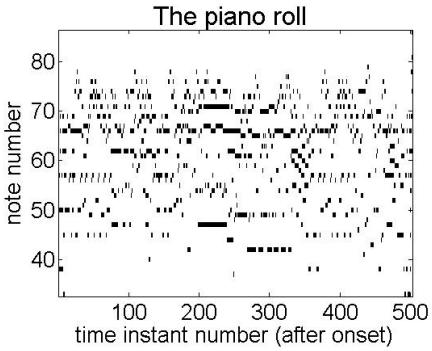


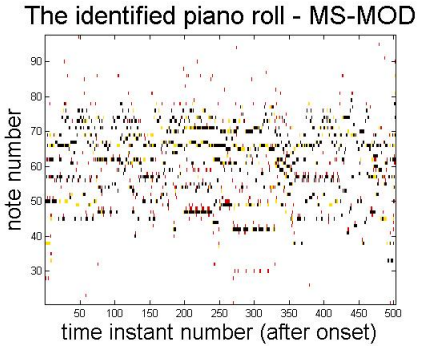
Accuracy=69.6%



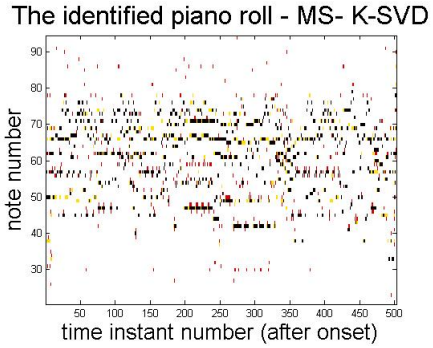
Accuracy=67.7%

- Complicated polyphonic music





Accuracy=64.0%



Accuracy=64.5%

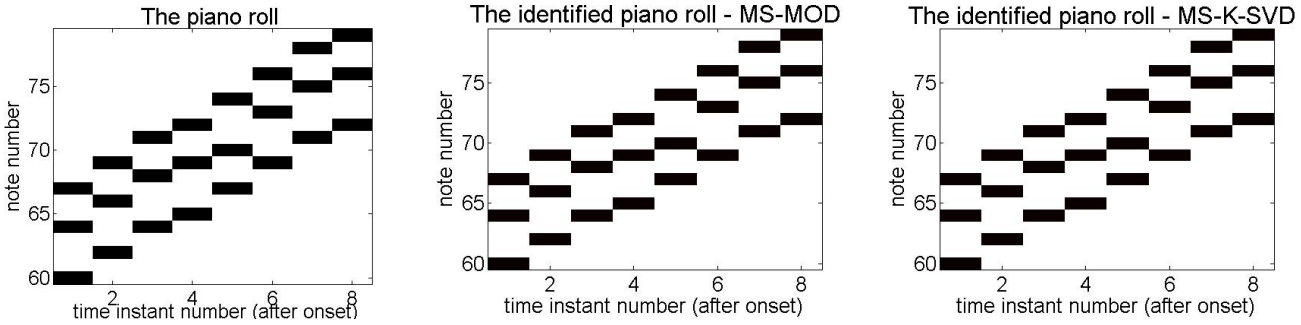
Synthesized MIDI music - a comparison to using an explicit dictionary

Accuracy (%)	MOD	KSVD	MS-MOD	MS-K-SVD
Monophonic music	100	100	100	100
Simple polyphonic music	37.6	39.5	69.6	67.7
Complicated polyphonic music	43.5	42.7	64.0	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.

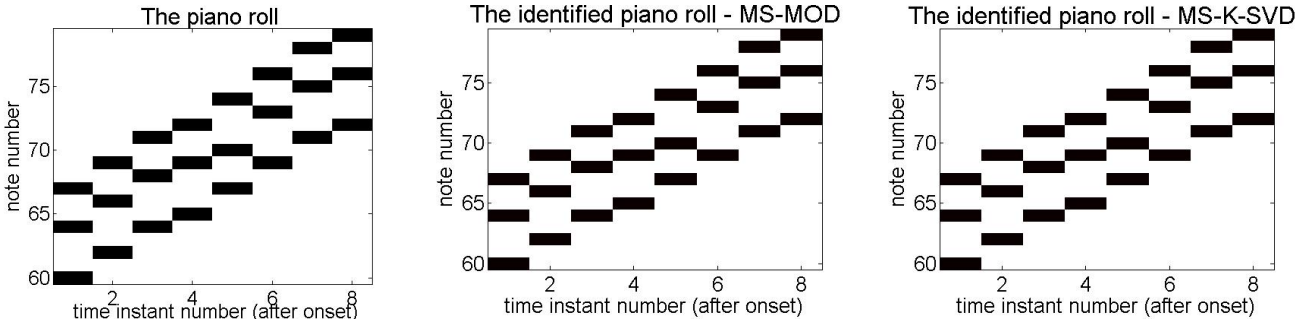
Synthesized MIDI music - a comparison to using an explicit dictionary

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

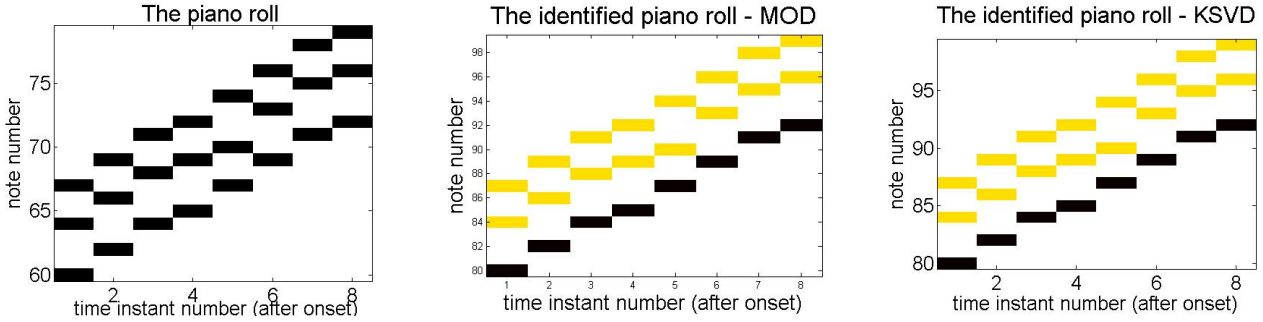


Synthesized MIDI music - a comparison to using an explicit dictionary

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



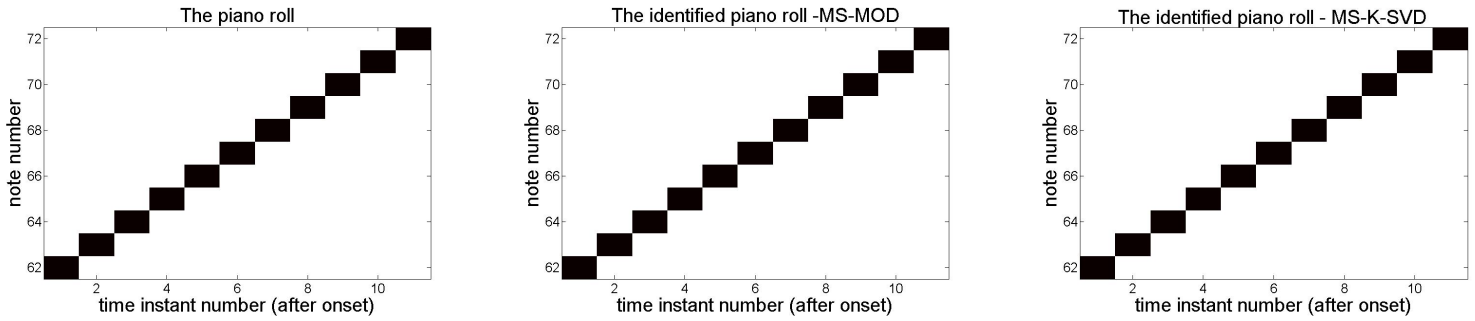
- Accords detection - MOD and K-SVD



- black = true positive, red = false positive, yellow = false negative.

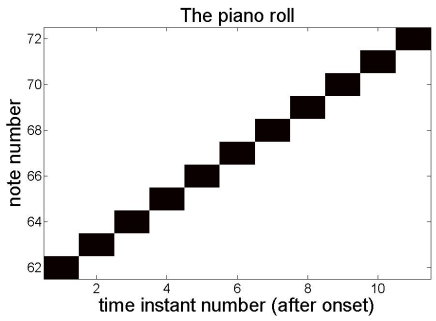
Recorded piano music - a comparison to using an explicit dictionary

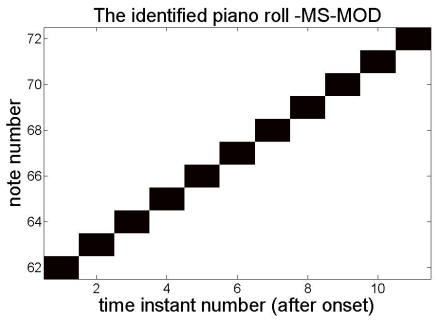
- Monophonic detection - MS-MOD and MS-K-SVD

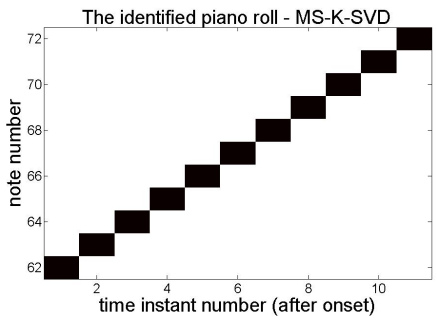


Recorded piano music - a comparison to using an explicit dictionary

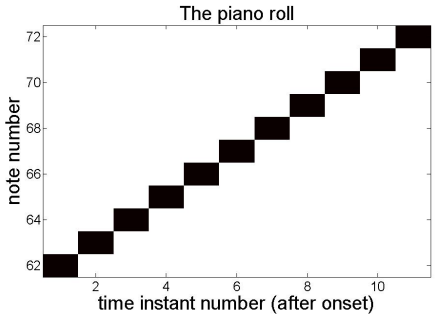
- Monophonic detection - MS-MOD and MS-K-SVD

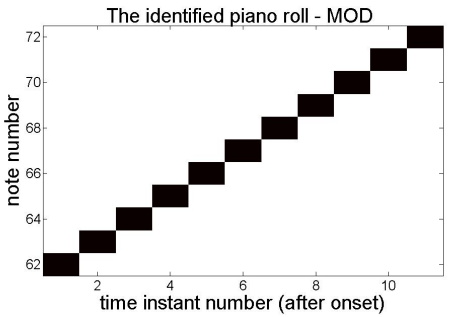


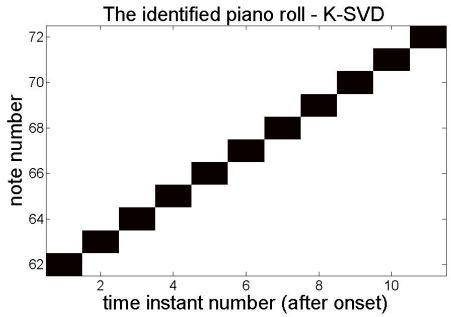




- Monophonic detection - MOD and K-SVD

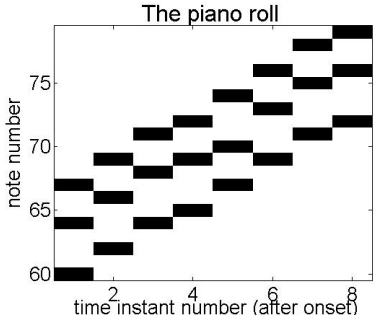


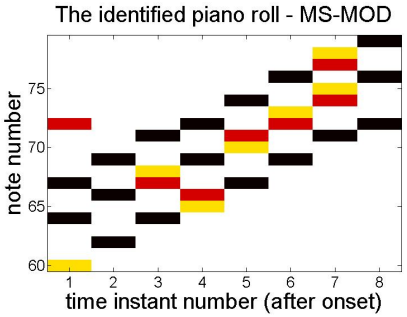




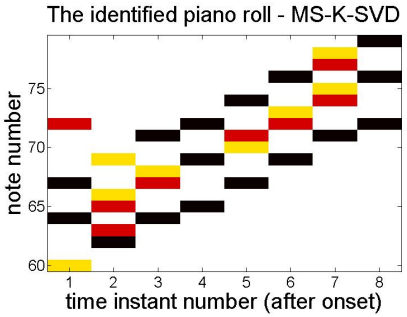
Recorded piano music - a comparison to using an explicit dictionary

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





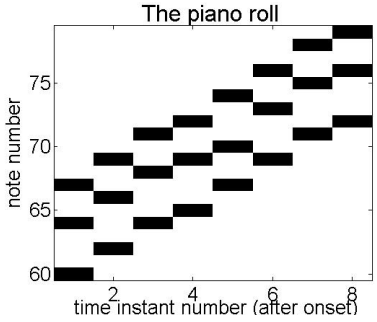
Accuracy=54.8%

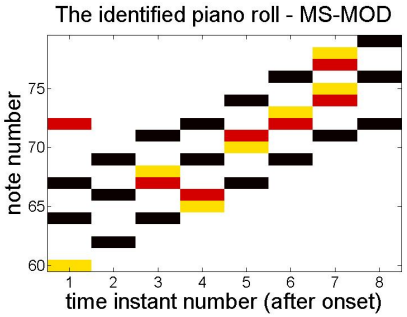


Accuracy=50.0%

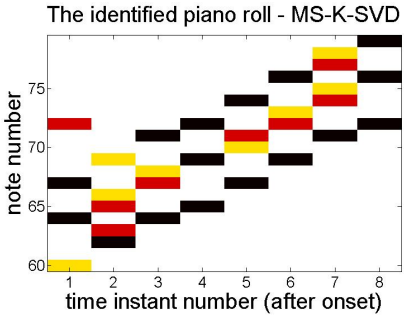
Recorded piano music - a comparison to using an explicit dictionary

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



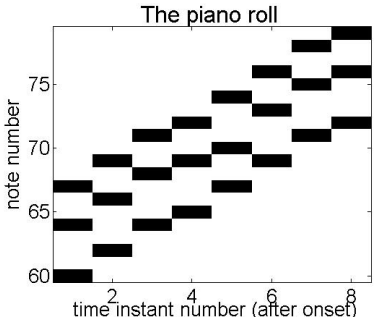


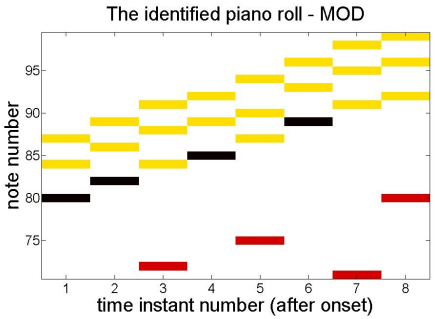
Accuracy=54.8%



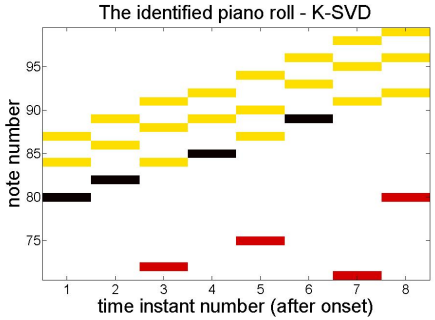
Accuracy=50.0%

- 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

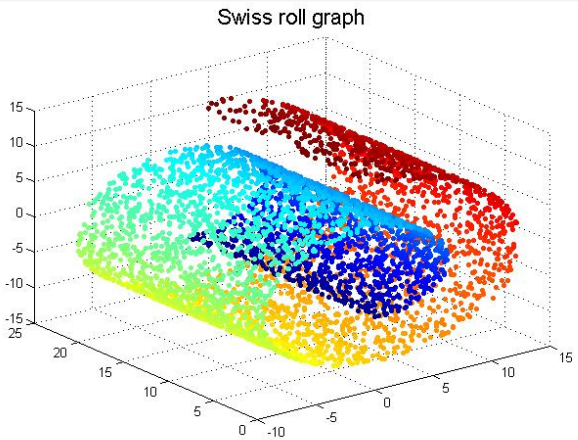
The feature vectors of natural audio and speech data lie on a non-linear manifold.

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]

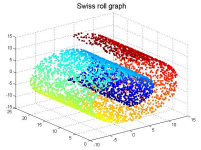
Manifold learning

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

1. Feature extraction
2. Manifold learning (Diffusion maps)
3. Classification

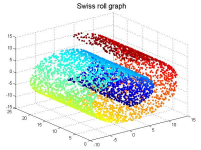
Diffusion maps - an overview

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



Diffusion maps - an overview

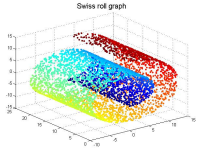
1. Build a connected graph from the feature vectors.
   - The nodes of the graph are the feature vectors $\{\mathbf{x}_i\}_i$.
   - The connections between the nodes are weighted by a symmetric, pointwise nonnegative kernel.



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

Diffusion maps - an overview

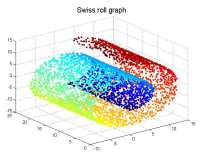
1. Build a connected graph from the feature vectors.
   - The nodes of the graph are the feature vectors $\{\mathbf{x}_i\}_i$.
   - The connections between the nodes are weighted by a symmetric, pointwise nonnegative kernel.



2. Define a Markov random walk on the graph and a right stochastic matrix describing its transitions.
3. Define the *Diffusion distances* $D_t(\mathbf{x}_i, \mathbf{x}_j)$ 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 $\{\mathbf{x}_i\}_i$.
   - 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* $D_t(\mathbf{x}_i, \mathbf{x}_j)$ as representing the connectivity in the graph .
4. Apply spectral decomposition of the right stochastic matrix, and derive the *Diffusion map* $\Psi_t(\mathbf{x}_i)$ of the feature vector $\mathbf{x}_i$.

Diffusion maps - main result

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

$$D_t(\mathbf{x}_i, \mathbf{x}_j) \simeq \|\Psi_t(\mathbf{x}_i) - \Psi_t(\mathbf{x}_j)\|_2$$

Classification of audio data using diffusion maps

- We can now use the diffusion maps $\{\Psi_t(\mathbf{x}_i)\}_i$ for classification instead of using the feature vectors themselves.

Classification of audio data using diffusion maps

- We can now use the diffusion maps $\{\Psi_t(\mathbf{x}_i)\}_i$ 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 of audio data using diffusion maps

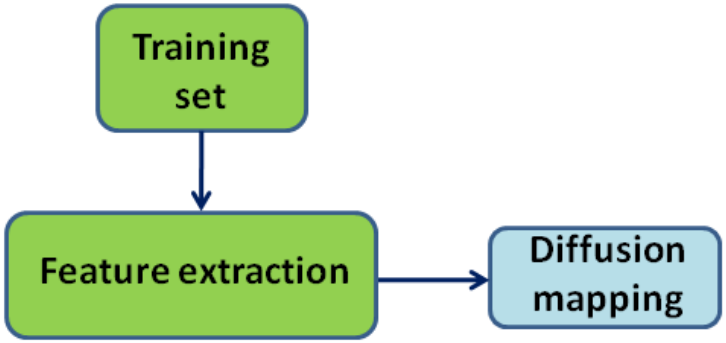
- We can now use the diffusion maps $\{\Psi_t(\mathbf{x}_i)\}_i$ for classification instead of using the feature vectors themselves.
- The Euclidean distances between the diffusion maps approximates the diffusion distances between the feature vectors
  $\Rightarrow$ *Classification based on Euclidean distances is meaningful*.

Classification of audio data using diffusion maps

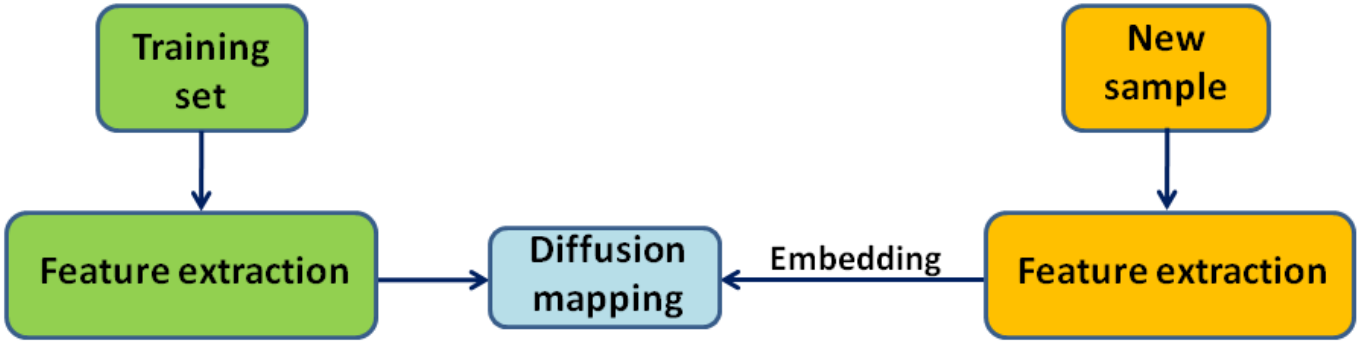
- We can now use the diffusion maps $\{\Psi_t(\mathbf{x}_i)\}_i$ for classification instead of using the feature vectors themselves.
- The Euclidean distances between the diffusion maps approximates the diffusion distances between the feature vectors
  $\Rightarrow$ *Classification based on Euclidean 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

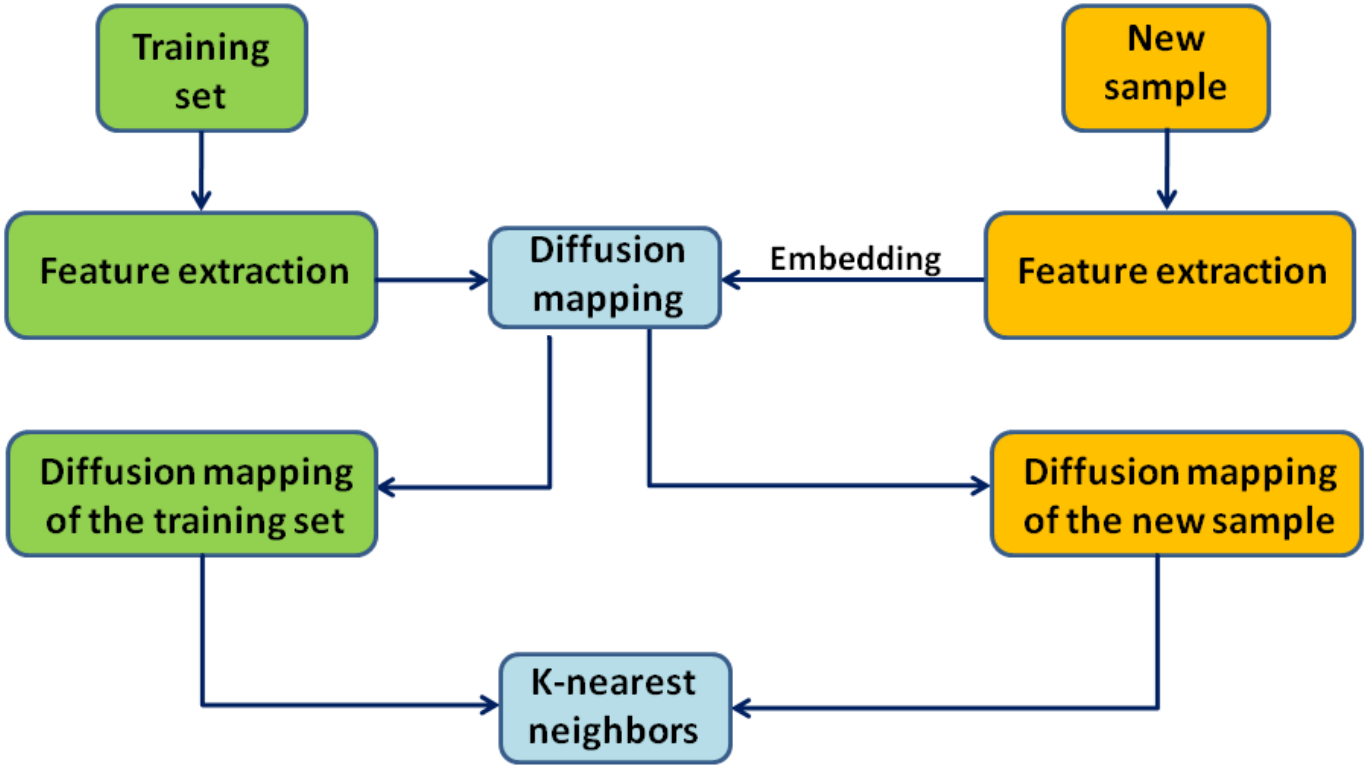
The experimental framework



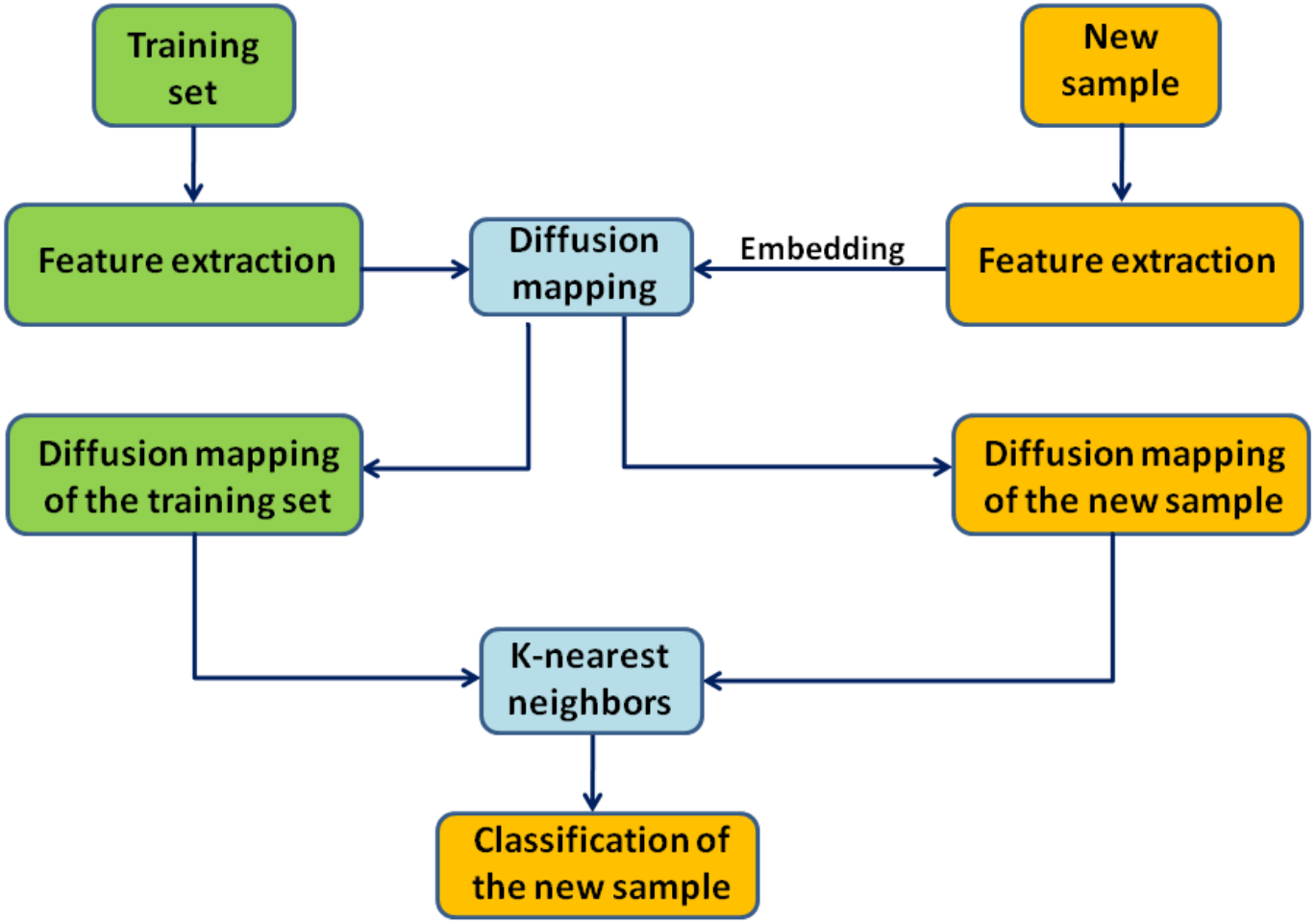
The experimental framework



The experimental framework



The experimental framework



Musical genre classification [Genussov & Cohen, 2010b]

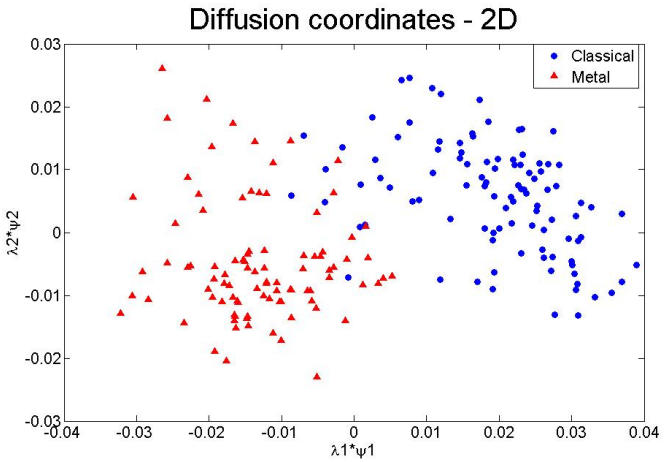
- GTZAN genre data set (`http://marsyas.info/download/data_sets`) - includes 1000 songs, 100 from each genre (Jazz, Classic, Metal, Rock, etc.) - divided randomly into training set and testing set.

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.

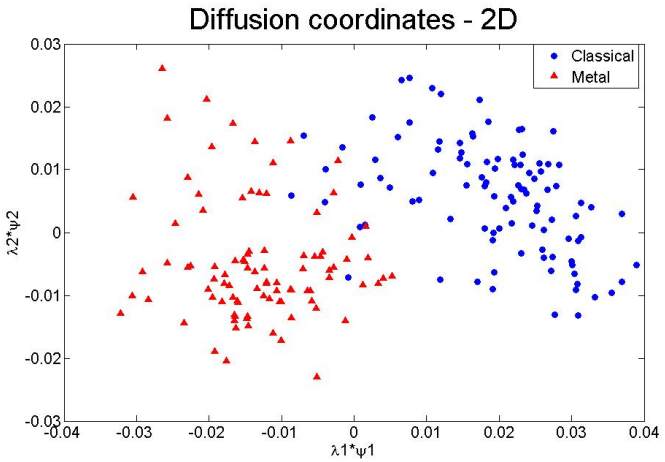
Musical genre classification [Genussov & Cohen, 2010b]

- 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:



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:



Musical genre classification - The confusion matrix

	"Blues"	"Classic"	"Country"	"Disco"	"Hiphop"	"Jazz"	"Metal"	"Pop"	"Reggae"	"Rock"
Blues	**0.60**	0.00	0.09	0.04	0.05	0.02	0.07	0.00	0.04	0.10
Classic	0.01	**0.81**	0.05	0.01	0.00	0.08	0.01	0.00	0.00	0.03
Country	0.07	0.02	**0.48**	0.05	0.01	0.11	0.00	0.06	0.07	0.13
Disco	0.02	0.00	0.08	**0.37**	0.11	0.03	0.07	0.12	0.06	0.14
Hiphop	0.04	0.00	0.02	0.08	**0.52**	0.00	0.02	0.13	0.16	0.02
Jazz	0.06	0.12	0.10	0.04	0.00	**0.54**	0.01	0.04	0.02	0.07
Metal	0.06	0.00	0.01	0.08	0.02	0.00	**0.73**	0.00	0.01	0.10
Pop	0.00	0.00	0.07	0.08	0.06	0.02	0.00	**0.68**	0.08	0.03
Reggae	0.05	0.00	0.07	0.05	0.07	0.01	0.01	0.07	**0.62**	0.05
Rock	0.06	0.00	0.19	0.09	0.04	0.10	0.11	0.05	0.05	**0.30**

Musical genre classification - comparison to other dimensionality reduction methods

	Dimensionality reduction type				
Number of classes	No dimensionality reduction	PCA	**Diffusion Maps**	Laplacian Eigenmaps	LLE
10	61.9%	60.3%	**56.3%**	51.7%	55.8%
5	87.5%	$86.7 \pm 4.5\%$	**83.4%**	83.2%	85.5%
2	98.4%	97.8%	**97.5%**	91.0%	96.3%
5 groups of pairs	69.1%	67.4%	**65.7%**	60.3%	65.7%

Unvoiced fricative phonemes identification [Genussov, Lavner & Cohen, 2010]

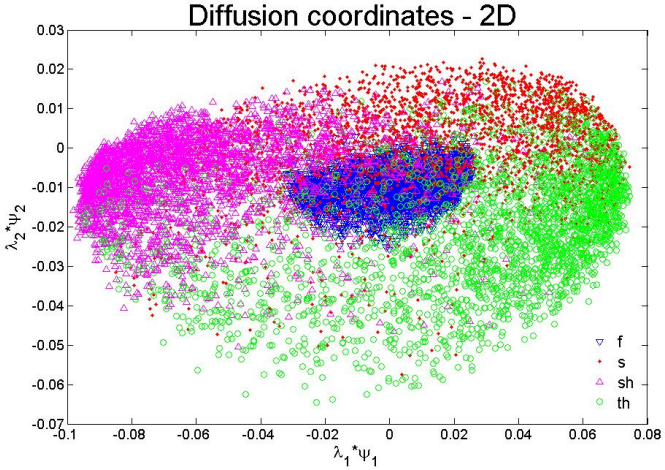
- The phonemes /s/, /sh/, /f/, /th/ (254 for each kind), from TIMIT data base - divided randomly into training set and testing set.

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.

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.
- For visualization, diffusion mapping to 2-D:



Unvoiced fricative phonemes identification - the confusion matrix

Table: Classification of phonemes - majority based

	"[f]"	"[s]"	"[sh]"	"[th]"
[f]	**0.87**	0.01	0.03	0.09
[s]	0.02	**0.87**	0.11	0.00
[sh]	0.06	0.09	**0.85**	0.00
[th]	0.27	0.06	0.01	**0.66**

Table: Classification of segments

	"[f]"	"[s]"	"[sh]"	"[th]"
[f]	**0.70**	0.04	0.06	0.20
[s]	0.07	**0.68**	0.16	0.09
[sh]	0.10	0.09	**0.79**	0.02
[th]	0.34	0.15	0.04	**0.47**

Unvoiced fricative phonemes identification - Comparison between dimensionality reduction methods

	Dimensionality reduction type		
	No dimensionality reduction	**Diffusion maps**	PCA
Classification of segments	68.8%	**66.6%**	68.6%
Classification of phonemes - majority-based decision	81.0%	**78.6%**	81.0%

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.

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

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


THANK
YOU