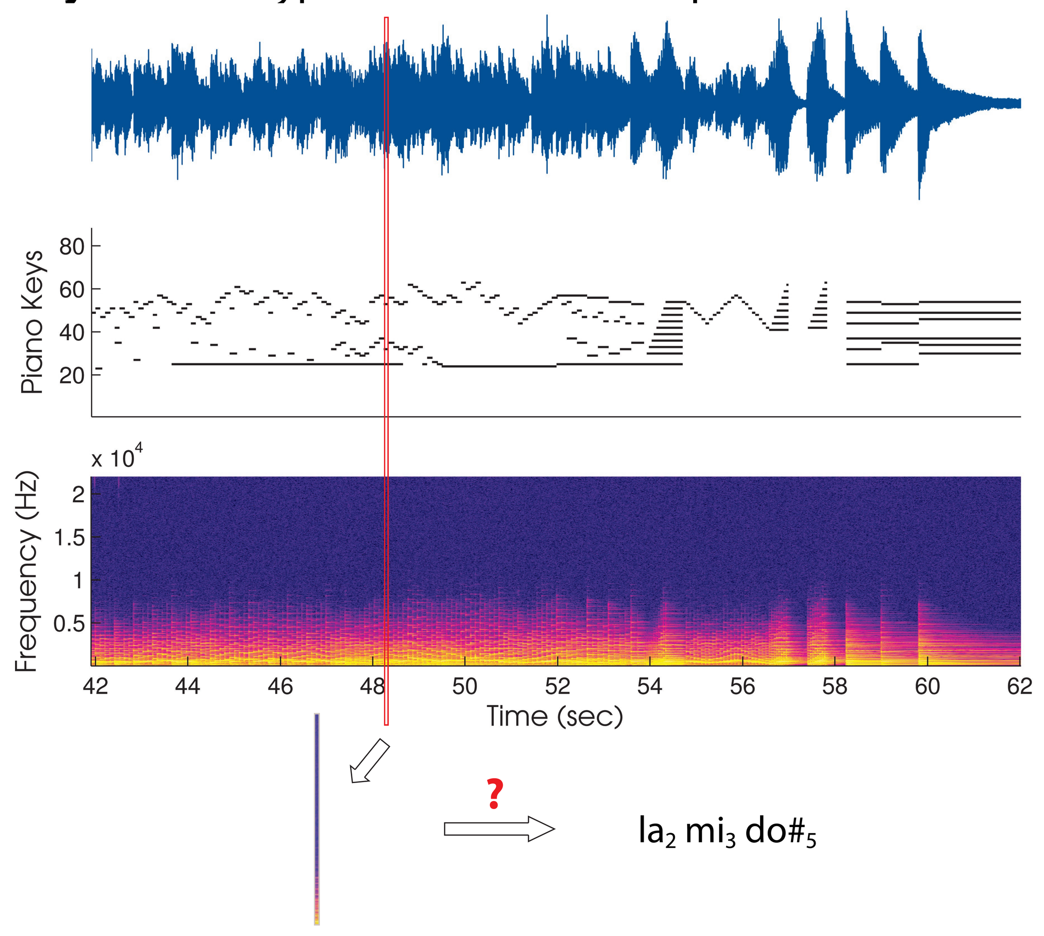
Large Scale Polyphonic Music Transcription Using Randomized Matrix Decompositions

İsmail Arı, Umut Şimşekli, Ali Taylan Cemgil, Lale Akarun

Dept. of Computer Engineering, Bogazici University, TURKEY

Introduction

Objective: Polyphonic music transcription

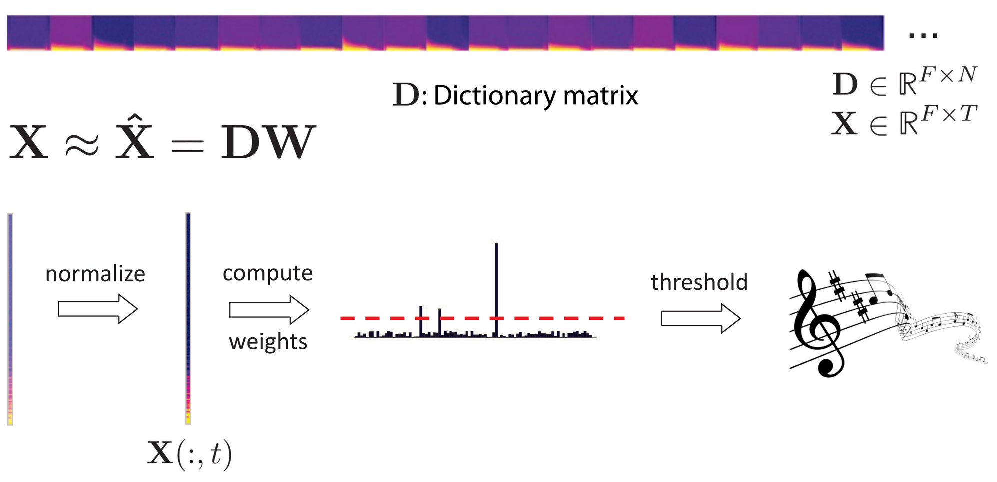


Requirements

- Fully automatic
- Deal with large data
- Efficiency (time & space complexities)
- Accuracy, simplicity and robustness

Materials & Methods

Linear model approach



$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{DW}$, $\mathbf{D} \in \mathbb{R}^{F\times N}$, $\mathbf{X} \in \mathbb{R}^{F\times T}$, $\mathbf{X}(:,t)$

Find $\mathbf{W}$ that minimizes the cost $\mathcal{D}[\mathbf{X}\|\mathbf{DW}]$.

We choose KL-divergence as cost function:

i. randomly initialize $\mathbf{W}$

ii. continue with the following step until convergence

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\mathbf{D}^\top \frac{\mathbf{X}}{(\mathbf{DW})}}{\mathbf{D}^\top \mathbb{1}} \right)$$

Division is done element-wise

$\odot$ implies element-wise (Hadamard) multiplication

$\mathbb{1}$ is a vector of ones of approppriate size

$\mathbf{D}^\top$ is the transpose of $\mathbf{D}$

☺ Simple model; easy to understand and implement

☹ Dictionary is huge in real apps; it may not fit into the RAM

☹ Matrix-products in the equation are costly

Improving efficiency via SVD

Use reduced SVD to compute rank-k approximation of $\mathbf{D}$.
Employ **randomized SVD** do deal with large data.

$$\underset{\tilde{\mathbf{D}},\ \mathrm{rank}(\tilde{\mathbf{D}})\le k}{\arg\min} \|\mathbf{D} - \tilde{\mathbf{D}}\|_F = \mathbf{A}_k \boldsymbol{\Sigma}_k \mathbf{B}_k^\top$$

Store $\mathbf{A}_k$ and store $\boldsymbol{\Sigma}_k\mathbf{B}_k^\top$ as $\tilde{\mathbf{B}}_k^\top$.

Modify the update rule:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\tilde{\mathbf{B}}_k \left(\mathbf{A}_k^\top \frac{\mathbf{X}}{(\mathbf{A}_k(\tilde{\mathbf{B}}_k^\top \mathbf{W}))} \right)}{\tilde{\mathbf{B}}_k(\mathbf{A}_k^\top \mathbb{1})} \right)$$

☺ Best factorization in terms of reconstruction error

☹ Singular vectors may not represent *physical reality*

Improving efficiency via CUR

Use $\mathbf{D} \approx \mathbf{CUR}$ to represent dictionary via its columns/rows.
$k_{\mathrm{col}}/k_{\mathrm{row}}$ are the number of selected columns/rows.
First, compute singular vectors, then

⇒ Probability of selecting i^{th} row is

$$\rho_i = \frac{1}{k}\sum_{j=1}^{k} A_k(i,j)^2, \quad i = 1,\dots,F$$

⇒ Probability of selecting j^{th} column is

$$\pi_j = \frac{1}{k}\sum_{i=1}^{k} B_k(j,i)^2, \quad j = 1,\dots,N$$

Select $k_{\mathrm{col}}/k_{\mathrm{row}}$ columns/rows randomly using a multinomial distribution with the computed probabilities.

Store $\mathbf{R}$ and store $\mathbf{CU}$ as $\tilde{\mathbf{C}}$.

Modify the update rule:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\mathbf{R}^\top \left(\tilde{\mathbf{C}}^\top \frac{\mathbf{X}}{(\tilde{\mathbf{C}}(\mathbf{RW}))} \right)}{\mathbf{R}^\top(\tilde{\mathbf{C}}^\top \mathbb{1})} \right)$$

☺ More interpretable decomposition

☹ Still trying to approximate the large dictionary matrix

Improving efficiency via selection of important data

Instead of using low rank approximation of the full matrix

⇒ Use only the selected columns

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\mathbf{C}^\top \frac{\mathbf{X}}{(\mathbf{CW})}}{\mathbf{C}^\top \mathbb{1}} \right)$$

⇒ Let $\check{\mathbf{D}}$ be the truncated matrix involving only the selected indices and $\check{\mathbf{X}}$ involve the selected rows of the input

Then, use only the selected indices

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\check{\mathbf{D}}^\top \frac{\check{\mathbf{X}}}{(\check{\mathbf{D}}\mathbf{W})}}{\check{\mathbf{D}}^\top \mathbb{1}} \right)$$

☺ Use only relevant information

☺ Fast & low space requirement

Time & Space Complexities *(in Big-O notation)*

	Time	Space
Full model	FNT	FN
SVD-based	$(F+N)kT$	$(F+N)k$
CUR-based	$(F+N)\min\{k_{\mathrm{row}},k_{\mathrm{col}}\}T$	$(F+N)\min\{k_{\mathrm{row}},k_{\mathrm{col}}\}$
C-based	$F k_{\mathrm{col}} T$	$F k_{\mathrm{col}}$
Skeleton	$k_{\mathrm{row}}\, k_{\mathrm{col}}\, T$	$k_{\mathrm{row}}\, k_{\mathrm{col}}$

Experimental Setup

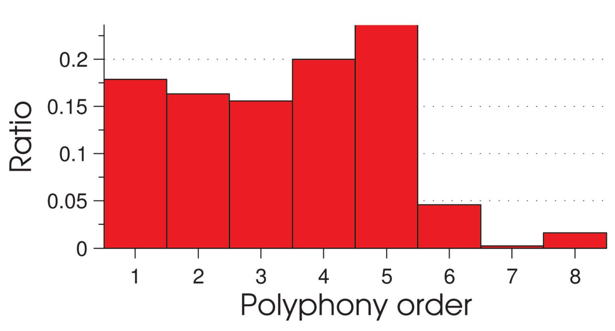
MAPS (MIDI Aligned Piano Sounds) Dataset

⇒ Training set

- 440 monophonic piano recordings
- Dictionary of size 1025 × 115600 (~ 860 MB)

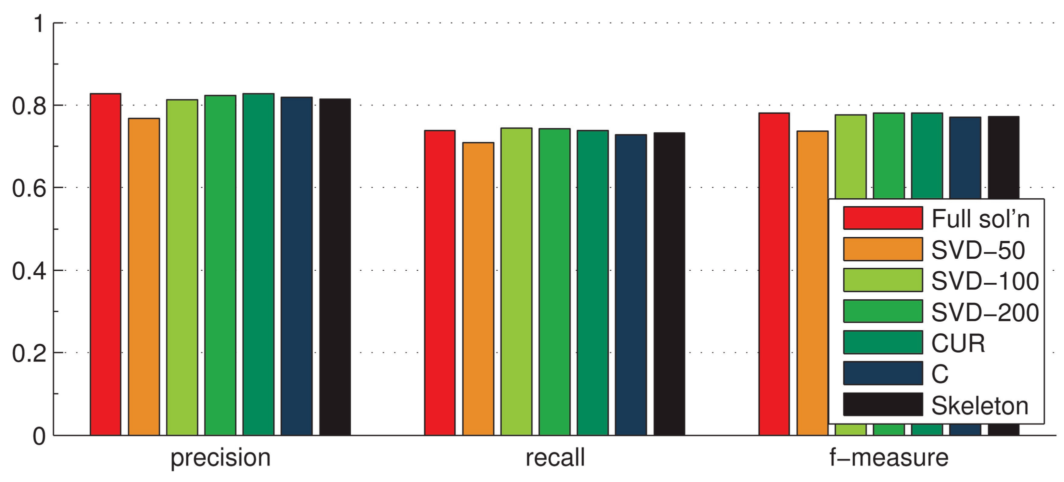
⇒ Test set

- Random sections from 5 different polyphonic pieces
- 3000 samples



Results

Results per approach



Precision, recall and f-measure per approach

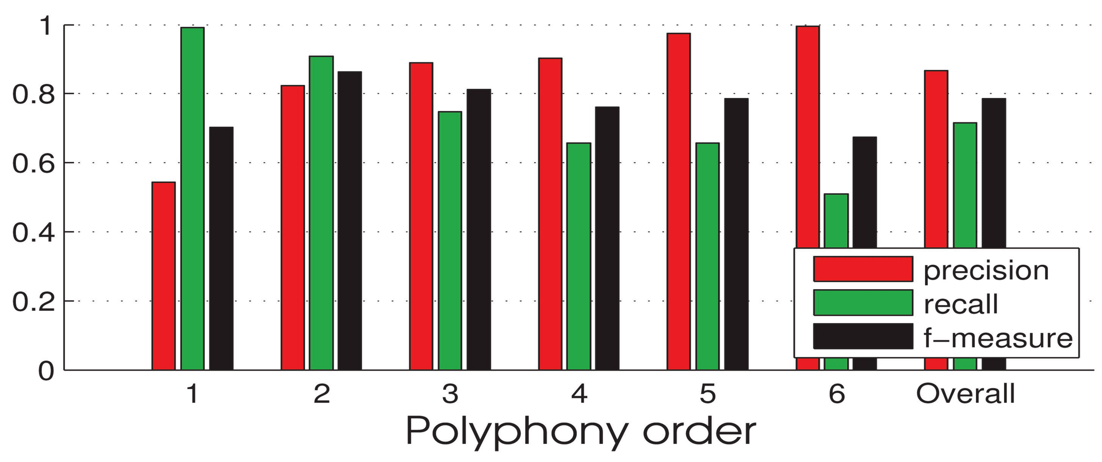
☺ For SVD-based approach, 200 dimensions (with 78.14% f-measure) is enough to maintain the original f-measure score (78.07%).

☺ We select 400 frequency bands and 4000 samples; CUR is a good alternative to SVD with similar results.

☺ C-based approach and skeletonizing (reducing both the #cols and the #rows) give promising results.

☺ We only use less than 2% of the data after skeletonization, and get an f-measure of 77.17%.

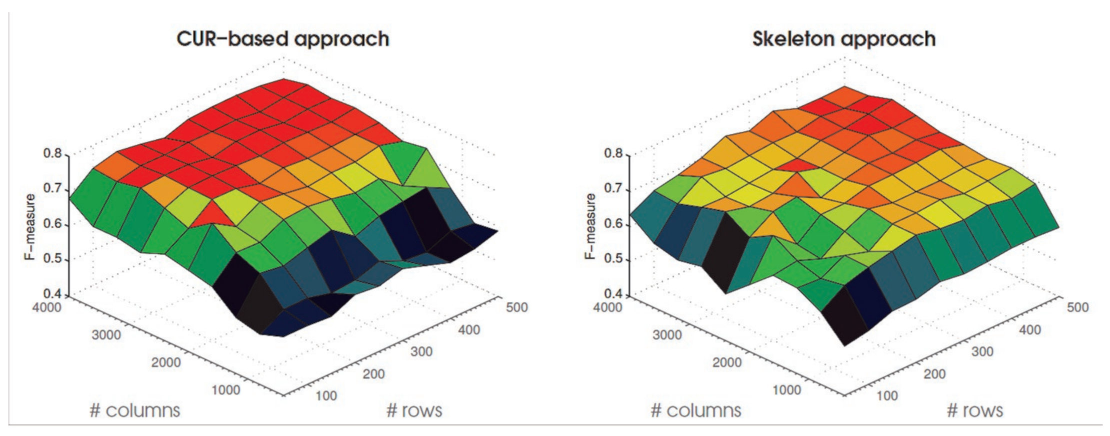
Results per polyphony order



Precision, recall and f-measure per polyphony order using skeletonization

☺ More than 65% f-measure for each polyphony order ⇨ Robust

Effect of #cols/#rows on the result

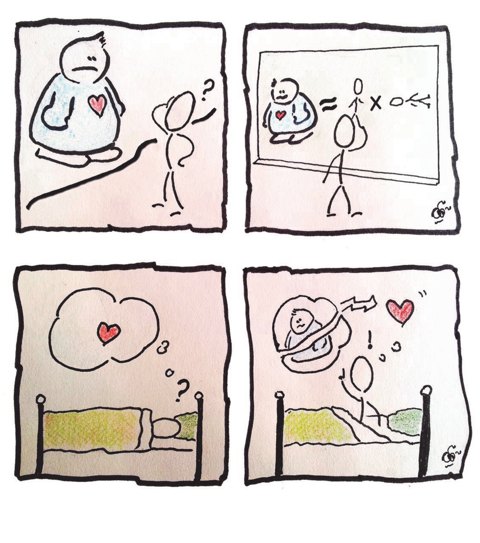


f-measures versus the #cols (samples) and the #rows (frequency bins)

☺ Only a few hundred frequency bins and a few thousand samples are sufficient to keep success ratio of the algorithm

Conclusions

- We show that even a standard matrix factorization model is prohibitive in real applications where a huge amount of data is used.
- A high f-measure value (~78%) is obtained on polyphonic recordings by using only a few hundred frequency bins and a few thousand sample columns out of a huge dictionary.
- The technique is simple, yet powerful and robust.
- Randomized matrix decompositions are crucial for practical issues.
- With abundance of data in different applications, randomized matrix decompositions are likely to get more attention in the future.



***Contact:** ismailar@boun.edu.tr*