Project report -Automatic Music Summarization via Similarity Analysis

Roland Nzala-Backa (École Polytechnique)

1 Abstract

Automatic segmentation and summarization of music is a key area of interest for music browsing, searching and representation. In this project, we will present and use a method developed in the paper "Automatic Music Summarization via Similarity Analysis" [1] by Matthew Cooper and Jonathan Foote. This method is based on similarity analysis : we do a parametrization of the audio of the music based on a window and we calculate between two pair of windows the the products that are embedded in a two dimension matrix. Then by studying a summation on columns, we find the similarity segment that best represents the song. For this project, we will first present with great details the method used and then apply it to different kind of music to analyze the results. We will expand the result of paper [1] by adding results of the euclidean distance and by using decreasing weights that favors the first part of the song. We will then add some limits of the paper.

2 Introduction

Audio signal processing is a key area of study for electronic treatment of sound waves. In this project, we will focus on the objective of summarization of music: the process of creating an accurate and condensed portrayal of a music piece.
This topic is widely use. Indeed, to present a user with a more representative part of a song, it useful to select a part of the music that is singular to the song.
It raises several questions. The question of parametrisation of the song is a key issue: we need to have a correct signal of the song that. We will use **Waveform Audio File Format files**, a format that stores audio bitstreams on PCs and some python libraries. Then we need a way to calculate the similarity between to frame of the signal: that is the role of similarity matrices.
All of this will be useful to explain the principle of summarization and use it later.

3 Self-similarity analysis

3.1 Parametrisation and Python

For our analysis we have audio sources from youtube videos. The source audio are on stereo format, a difference from [1] that helps us to have a more accurate representation of the song. These are transformed into **Wave Audio File Format** (WAV). WAV is an audio file format for compressed audios that samples the audio at 44.1 Hz. The audio are parameterised with 100 ms time frame using **Mel-Frequency Cepstral Coefficient** (MFCC). The MFCC parametriation of an audio signal is a small set of features that described the overall shape of the signal enveloppe, and the coefficients are computed with Fourier linear cosine transform of the log power spectrum. The result for each frame is a compact vectors of size 13.

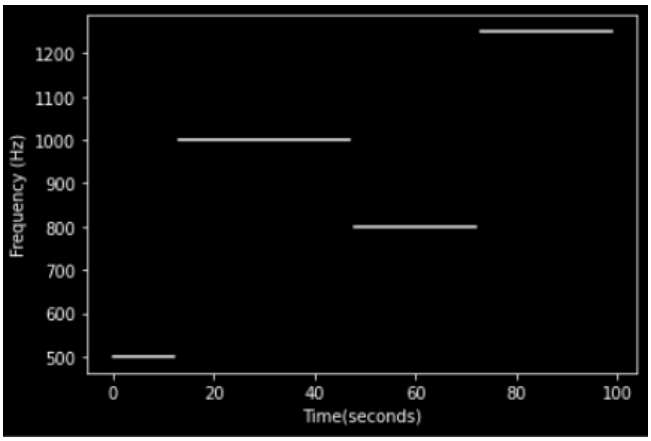

Figure 1: Spectrogram data of synthetic signal

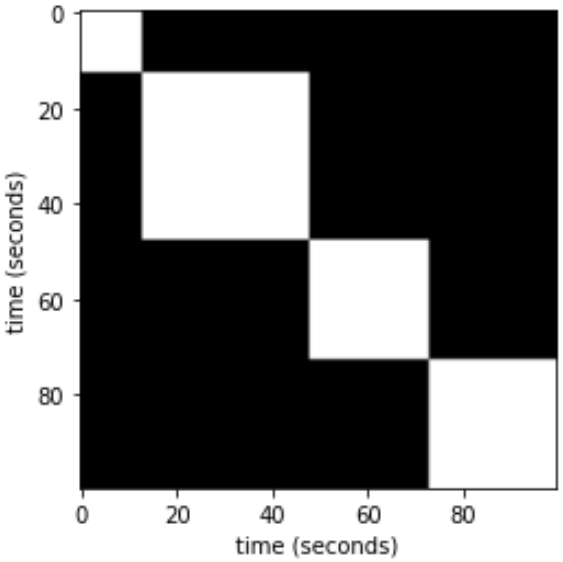

Figure 2: Similarity matrix of synthetic signal of figure 1

Figure 1 represents an artificial spectrum of a synthetic data where inputs are in white. This is an artificial example that will be useful to better understand the context of our analysis .

3.2 Distance Matrix Embedding

In this section. Once the signal is parameterised, we want to represent it in a matrix. The main objective is to measure the similarity of two vectors v_i and v_j chosen at frames i and j, $i, j \in [[1 : N]]$ and N is the total number of frames. We can use several distances measures.

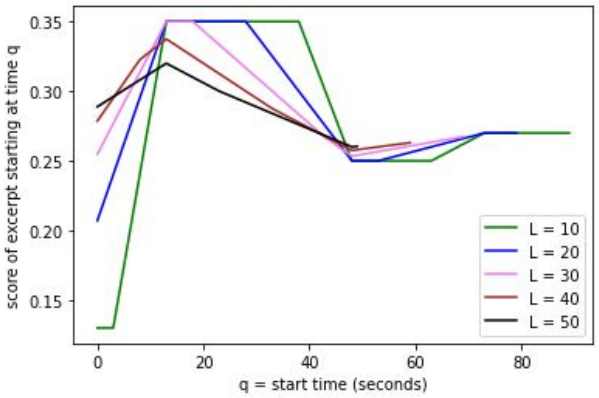

Figure 3: Summary scores $Q_L(i)$ computed from the similarity matrix of Figure 2

$$d_e(v_i, v_j) = \sqrt{\sum_{l=1}^{L}(v_i(l) - v_j(l))^2} \tag{1}$$

A first kind of measure (1) is the Euclidean distance of an L (here L=13) dimensional space. It is very useful we are able to see disparity for components.

$$d_c(v_i, v_j) = \frac{v_i.v_j}{\|v_i\|.\|v_j\|} \tag{2}$$

A second measure of distance (2) is the cosine distance of a space of dimension L (here L=13). It is very useful we are able to see similarities even if the norm is little, for example between two silences frames.

$$d_0(v_i, v_j) = 1_{v_i=v_j} \tag{3}$$

A third measure of distance (3) is the 0-1 distance. It is basic for comprehension but not useful for real data.
Then, in the matrix of similarity S, we compute all the distances between vectors representing each frame, when $d \in [d_0, d_c, d_e]$ and $i, j \in [[1 : N]]$

$$S(i, j) = d(v_i, v_j) \tag{4}$$

3.3 Visualizing Similarity Matrices

We can visualise the distance matrix to better understand the structure of the audio. Regions with high similarity are in white, regions with less similarity are in white. The diagonal by definition is in white because a frame is similar to itself. We use the 0-1 distance for our example and we obtain **Figure 2** (0 values in black, 1 values in white);

4 Principle of summarization

The exact motivation of the method of summarization is explain in [1].
The similarity of a segment between times q and r is calculated as the sum of the self-similarity between the segment and the entire work normalized by the segment length. We have the following formula where N is the length of the entire work. We finally have a similarity matrix $S_{average}$ of size $N \times N$, and its coefficients are

$$S_{average}(q, r) = \frac{1}{N(r-q)} \sum_{m=q}^{r} \sum_{n=1}^{N} S(m, n) \tag{5}$$

We will also use that weight functions that decreases on time to favors first parts of the song.

$$S_{average,w}(q, r) = \frac{1}{N(r-q)} \sum_{m=q}^{r} \sum_{n=1}^{N} w(n) S(m, n) \tag{6}$$

To optimal summary of length L and of weight w, we just find the maximal value of all $S_{average,w}(q, r)$ for $q, r \in [[1 : N]]$ such that $r - q = L$. The score $Q_{L,w}(i)$ is computed as follows.

$$Q_{L,w}(i) = S_{average,w}(i, i + L) = \frac{1}{NL} \sum_{m=i}^{i+L} \sum_{n=1}^{N} w(n) S(m, n) \tag{7}$$

The **Figure 3** shows all the value of $Q_L(i)$ for the synthetic signal for different values for a start time at q.
Then to have the best summary of size L, we only to find the maximum value of $Q_{L,w}(i)$

$$q_{L,w} = Argmax_{1 \le i \le N-L} Q_{L,w}(i) \tag{8}$$

In **Figure 3**, we can see that for $L \in [10, 20, 30, 40, 50]$, at least one of the best summary starts at time $q = 13s$

5 Experiments

5.1 Algorithms and implementation

All of our code is in Python. The implementation of the method presented in [1] is straightforward.
We will nevertheless use this algorithm to compute the score based on the similarity matrix S of size N with a complexity of $O(N^2)$ instead of the naive method of complexity $O(NL(N - L))$. The last complexity is not good: if we want to have a representation with 10% of the song for example, the last complexity is proportional to $O(N^3)$.

Algorithm 1: Computation of score

Data: $\mathcal{S}, L$
Result: Q_L, q_L
$Q_L \leftarrow []$
$C_1, ..., C_N \leftarrow \frac{\sum_{n=1}^{N} S(1,n)}{N \times L}, ..., \frac{\sum_{n=1}^{N} S(N,n)}{N \times L}$
$Q_L(1) \leftarrow \sum_{m=1}^{L} C_m$
$q_L \leftarrow 1$
$max \leftarrow Q_L(1)$
for $i \leftarrow 1$ ***to*** $N - L - 1$ **do**
  $Q_L(i + 1) = Q_L(i) - C_i + C_{1+i+L}$
  if $max \le Q_L(i + 1)$ **then**
    $q_L \leftarrow i + 1$
  end
end

Because WAV files are sometimes heavy (41 Mo for *Ramenez la coupe à la maison*, 28 Mo for *God save the queen*, 47 Mo for *Wildest Dreams*, the data is large, and the computation of algorithms can take several minutes.

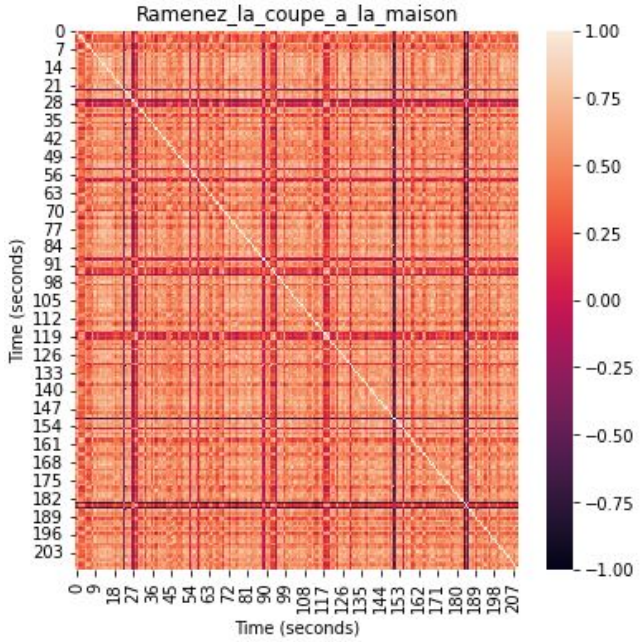

Figure 4: Similarity matrix computed for Vegedream's *Ramenez la coupe à la maison* using MFCC features and euclidean norm similarity measure

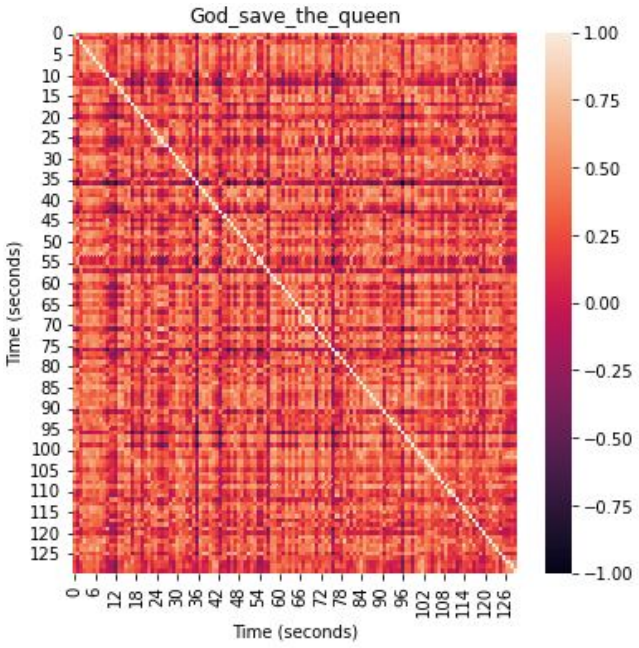

Figure 5: Similarity matrix computed for *God Save the Queen* using MFCC features and euclidean norm similarity measure

5.2 Dataset and Music Visualisation

We will study three different songs:
- *Ramenez la coupe a la maison* by Vegedream
- *God save the queen* (United Kingdom anthem)
- *Wildest Dreams* by Taylor Swift

We study piano (struck string instrument) versions of *Ramenez la coupe a la maison* and *God save the queen* that have higher amplitude of harmonics $O(\frac{1}{N})$ than violon (pinched string instrument with amplitudes of harmonics proportional to $O(\frac{1}{N^2})$, the version of *Wildest Dreams* studied Each audio at 44.1 Hz are parsed at 10Hz rate. We use Python package for MFCC. Contrary to [1], we use all coefficients of the MFCC to compute the similiraty matrix. We scaled the coefficient to a zero mean and a unite variance. We visualize for two distances, the euclidean distance (1) and the cosine distance (2).

In **Figure 4**, **Figure 5** and **Figure 6** we visualize the similarity matrix with euclidean distance. We scale the matrix such that the maximum value is 1, the minimum is -1.

In **Figure 7**, **Figure 8** and **Figure 9** we do the same thing with the cosine distance.

We can see that the cosine distance is more useful for this task and it helps us to separate more easily the data. Therefore it is the distance that we will use after in the project.
The texture type of **Figure 9** is really different from **Figure 7** and **Figure8** : this can be explain that contrary to other songs that are interpreted with piano, *Wildest Dreams* is played with violon here. With lower amplitudes harmonics for each fundamental frequency than piano, we see that the song similarity matrix has more differentiated areas, brighter or darker ones. For the song *Ramenez la coupe à la maison*, we clearly see bright squares that are repetitions of the theme of the song in several verse. For *God save the queen* we have a compact

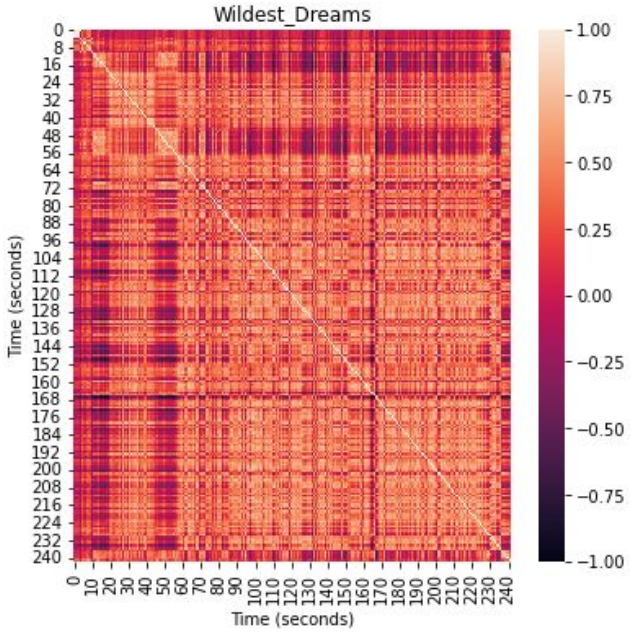

Figure 6: Similarity matrix computed for Taylor Swift's *Wildest Dreams* using MFCC features and euclidean norm similarity measure

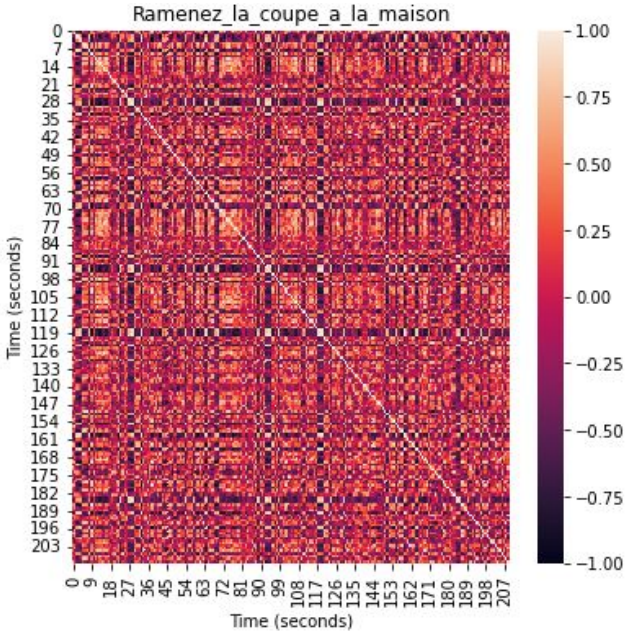

Figure 7: Similarity matrix computed for Vegedream's *Ramenez la coupe à la maison* using MFCC features and cosine similarity measure

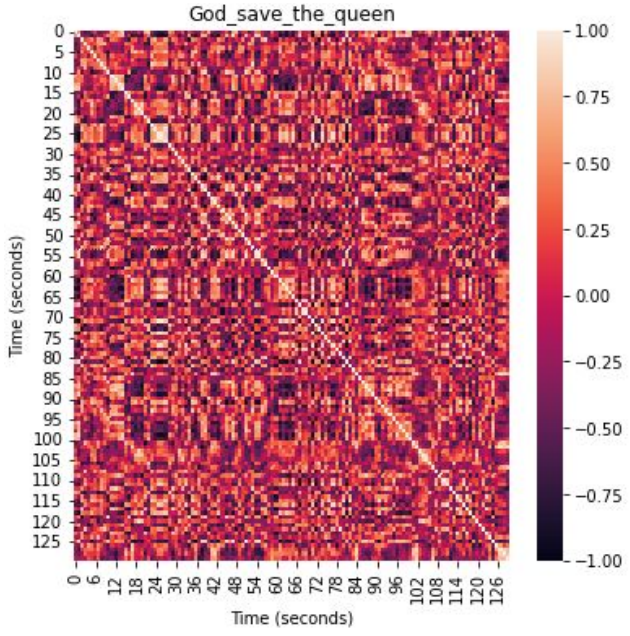

Figure 8: Similarity matrix computed for *God Save the Queen* using MFCC features and cosine similarity measure

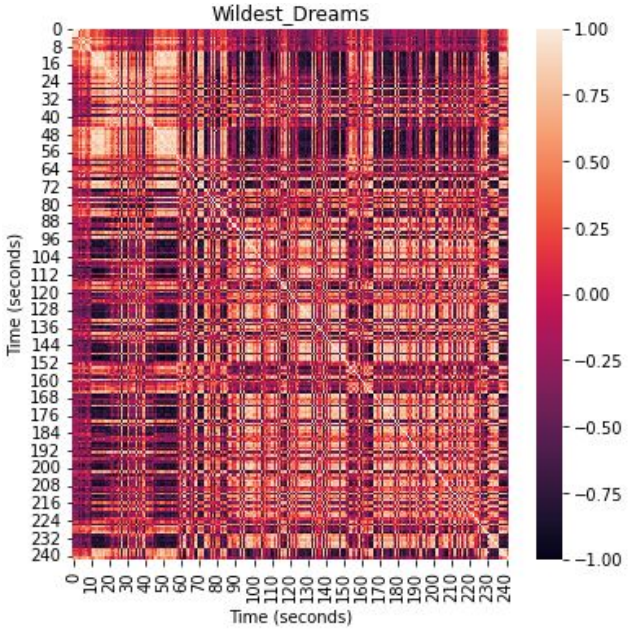

Figure 9: Similarity matrix computed for Taylor Swift's *Wildest Dreams* using MFCC features and cosine similarity measure

Segment length	Start (sec)	End (sec)
10	74.4	84
20	65.4	85.4
30	103.8	133.8

Table 1: Summary times for *Ramenez la coupe a la maison* with w_0

song, and for *Wildest Dreams* we observe a division between the start and the end of the song, with more visible differences.

5.3 Summary scores and times with different weighted measures

For the following experiments, we will now try to find the best summary times for segments of lengths 10, 20 and 30 seconds. We will try two different measures:
- the first one is the unite measure with for all non negative integers n, $w_0(n) = 1$. In that case, each part of the song as the same value.
-the second unite measure is an decreasing exponential function, where for all integers n, $w_\lambda(n) = exp(-\frac{\lambda n}{N})$, où $\lambda \in R^+$. The decreasing exponential function favors early parts of the song. The use is motivated because the amplitude of sound a measure usually in decibel (dB), a logarithmic scale.

We compute the summary scores for *Ramenez la coupe à la maison*.

We clearly see on **Figure 10** and **Figure 11**that the decreasing exponential function tends to diminish the score for last parts of the song, in similar proportion for every value of L.

Then we study the summary times of these three songs.

We have with w_0 the first results in **Table 1**, **Table 2**, **Table 3**. We see that the predicted summaries are not in the start of

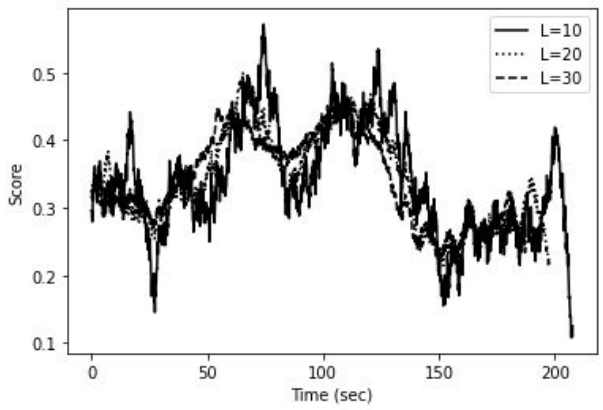

Figure 10: Summary scores $Q_L(i)$ computed from the similarity matrix of Figure 7

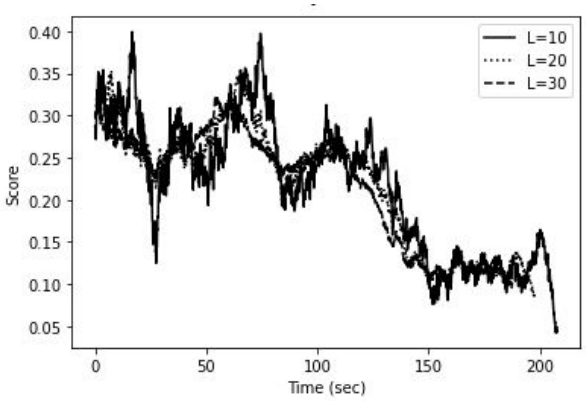

Figure 11: Summary scores $Q_L(i)$ computed from the similarity matrix with w_1

Segment length	Start (sec)	End (sec)
10	46.2	56.2
20	34.2	54.2
30	32.7	62.7

Table 2: Summary times for *God save the queen* with w_0

Segment length	Start (sec)	End (sec)
10	200.6	210.6
20	90.8	110.8
30	185.4	215.4

Table 3: Summary times for *Wildest Dreams* with w_0

Segment length	Start (sec)	End (sec)
10	16.6	26.6
20	65.5	85.4
30	54.3	84.3

Table 4: Summary times for *Ramenez la coupe a la maison* with w_1

Segment length	Start (sec)	End (sec)
10	4.4	14.4
20	4.4	24.4
30	32.7	62.7

Table 5: Summary times for *God save the queen* with w_1

the songs.

With greater details, we see that for *Ramenez la coupe à la maison*, optimal summaries are located at the third verse of the song with a rhythm that characterizes the song. For *God save the queen*, the most representative extract are in the first half, while for *Wildest Dreams*, the summary times are focused on the chorus.

We then have results with with w_1 ($\lambda = 1$ as scaling factor) in **Table 4**, **Table 5**, **Table 6**. The exponential term favors each time the start of the song.

6 Conclusion

We have made, as in [1], a quantitative approach to music summarization. The similarity between pairs of audio features is encapsulated within a similarity matrix, which unveils the primary structure of the audio. We conducted experiments We added some new tests, such that the effects of instruments and the effect of weighted average that favors first parts of the song. We finally obtained some interesting results that prove the efficiency of the method.

References

[1] M.Cooper and J.Foote, Automatic Music Summarization via Similarity Analysis, 2016

Roland Nzala-Backa [roland.nzala-backa@polytechnique.edu] is a student from the Mathematics, Vision & Learning (french : "Mathématiques, Vision, Apprentissage") Master of Science, created and driven by the mathematics department of École Normale Supérieure Paris-Saclay.

Segment length	Start (sec)	End (sec)
10	30.8	40.8
20	90.2	110.2
30	89.3	119.3

Table 6: Summary times for *Wildest Dreams* with w_1