LAB 5: MUSIC CLASSIFICATION

Author

Enric Verdaguer Morales
SMC Master
Universitat Pompeu Fabra

All code, documentation, feature tables and Audio Samples used in this lab can be found in the following link: https://github.com/enric592/MIR_Lab5

1. INTRODUCTION

In this lab we are going to build a music classifier, concretely we are going to work in mood. The general aim of our work is to learn about the whole process of building a classifier by working with a reduced dataset and taxonomy in order to keep it as simple as possible.

2. TAXONOMY DEFINITION

We decided to build a 'mood' classifier, by considering the most easy and well-known moods.
The given database for this lab had three different classes: Happy, Sad and Aggressive, but we took the decision to work only with the first two ones because we were looking for the most easy and clear case.
Additionally, while Happy and Sad are known as complete opposites, being easy for people to distinguish between them, aggressiveness distinction looks a little more vague to me:
For me 'Aggressive' it is not a completely independent entity: I can think about some cases of Happy and Aggressive songs, but I can not imagine any excerpt being Happy and Sad simultaneously.

3. DATASET DEFINITION

For training purposes we use the given Happy/Sad database composed by 60 samples: 34 of them are tagged 'Happy' and the other 26 are sad:

Song	Artist	Mood
Black Is The Color of My True Loves Hair	Nina Simone	Sad
Honey	Erykah Badu	Happy
Truth	Chiddy Bang	Happy
Good Life (Ft. T-Pain)	Kanye West	Happy
Todo el mundo lo sabe	Solo los Solo	Happy
Por que te fuiste?	MDE Click	Sad
Runaway (Ft. Pusha T)	Kanye West	Sad
Down In A Hole	Alice in Chains	Sad
Sweet Home Alabama	Lynyrd Skynyrd	Happy
Brothers In Arms	Dire Straits	Sad
Mouthful Of Cavities (24-Bit Digitally Remastered 01)	Blind Melon	Sad
Murir	MDE Click	Sad
Rainy Day Women # 12 & 35	Bob Dylan	Happy
Here Comes Your Man	Pixies	Happy
Troy (2009 Digital Remaster)	Sinéad O'Connor	Sad
Adagio in G minor	Albinoni	Sad
Adagio for Strings	Samuel Barber	Sad
Always	Bon Jovi	Sad
Fingerbib	Aphex Twin	Happy
Dance Little Liar	Arctic Monkeys	sad
Bang Bang	Nancy Sinatra	Sad
Bessarabian horadi sapozhkelekh	Kroke	Sad
Stir it up	Bob Marley	Happy
Brother Bru	Billy Martin & Wil Blades	Happy
Cloud Atlas End Title	Unknown	Sad
Collapse The Light Into The Earth	Porcupine Tree	Sad
Country House	Blur	Happy
Dance, dance	Fall out boy	Happy
Power Ballad	Datasette	Happy
New Beginning	Deaf Center	Happy
Nantes	Barbara	Sad
Don´t Cry	Guns´n Rose	Sad
Don't stop me now	Queen	Happy
Waternight	Eric Whitacre	Sad
Excuse me	Peter Gabriel	Happy
Jungle Drum	Emiliana Torríni	Happy
I'm into Something Good	Herman's Hermits	Happy
Surfin' USA	The Beach Boys	Happy
The Air-Conditioned Nightmare	Mr Bungle	Happy
Nude	Radiohead	Sad
I've Been Loving You Too Long	Otis Redding	Sad
Theme from Schindler's List	John Williams	Sad
The Air-Conditioned Nightmare	Mr Bungle	Sad
Gloomy Sunday	Rezso Seress	Sad
Goodbye My Lover	James Blunt	Sad
Goodbye to Romance	Ozzy Osbourne	Sad
Since I Left You	The Avalanches	Happy
Sexual Healing	Hot 8 Brass Band	Happy
We're from Barcelona	I'm from Barcelona	Happy
Anyone Else But You	The Moldy Peaches	Happy
Unknown	Unknown	Happy
High Hopes	Pink Floyd	Sad
How Long	Eagles	Happy
Hurt	Johnny Cash	Sad
Ice Cream Man	Van Halen	Happy
Ha en rozsa volnek	Janos Brody	Sad
Soda's Theme	Ilkae	Happy
Is this love	Bob Marley	Happy
Je Veux	Zaz	Happy
Restless	Kakkmaddafakka	happy
La Caravane	Caravane Place	Happy
Lacrimosa	Mozart	Sad
Lola's Mambo	Juan Luis Guerra	Happy
Long Live Rock 'n' Roll	Rainbow	Happy
La primavera Trompetera	Los delinqüentes	Happy
Love Today	Mika	Happy
Mamma Mia	ABBA	Happy
Red Paint	Matt and Kim	happy
Moonlight Sonata	Ludwig van Beethoven	Sad
Sunburn	Muse	sad
Neues Liebeslied	Kleinggeldprinzessin	Sad
Familiar	Nils Fraum	Sad
Nothing else matters	Metallica	Sad
A1	Olafur Arnalds	Sad
Once more once	Michel Camilo	Happy
Pirata Capitán	La Kinky Beat	Happy

Spring	Vivaldi	Happy
Karma Police	Radiohead	sad
Roads	Portishead	Sad
Pyramid Song	Radiohead	Sad
Untitled #1 (Vaka)	Sigur Rós	Sad
Pictures Of You	The Cure	Sad
Sacrificed Sons	Dream Theater	Sad
Legalizacion	Ska P	Happy
Still Got the Blues	Gary Moore	Sad
John Wayne Gacy Jr	Sufjan Stevens	Sad
Sunrise to sunset	Edna's Goldfish	Happy
Horny	Mousse T	Happy
Tears in Heaven	Eric Clapton	Sad
Zombie	The Cranberries	sad
Same Jeans	The View	happy
Tus Ojos	Guerola	Sad
Vacant	Dream theater	Sad
A-punk	Vampire Weekend	happy
Walking on Sunshine	Katrine and the Waves	Happy
When You Are Gone	Avril Lavingne	Sad
You Shook Me All Night Long	AC/DC	Happy
Young for You	Gala	Happy
I'm Waiting For The Man	The Velvet Underground / Nico	Happy

Table 1. Training dataset.

Apart from this given set we made our own contribution by adding 10 more excerpts. The idea is to use them as a test set to evaluate the built classifier with samples of known songs. This small evaluation set is composed by five samples of each instance with a duration of around 30 seconds (same as training).

Song	**Artist**	**Mood**
Soc feliç	Hora de Joglar	Happy
Waterloo	Abba	Happy
Warmness on the Soul	Avenged Sevenfold	Sad
Everybody Talk	Neon Trees	Happy
My Immortal	Evanescence	Sad
Abril 74	Lluís Llach	Sad
Happy	Pharell Williams	Happy
Love of my life	Queen	Sad
Stay with me	Sam Smith	Sad
Wake me up before you go go	Wham!	Happy

Table 2. Test dataset.

4. FEATURE EXTRACTION

We developed a feature extractor using Matlab and MIR toolbox, and exported our results into a .csv file.

As mood is something a little vague to define, and is not directly represented by any descriptor, we are going to use many of them that can be indirectly related in order to extract enough information to build a good classifier.

Considering that we are working in a quite small dataset of 60 instances for training, we have to use a reduced set of features to avoid over-fitting, so we are picking only those that make real sense to us and we can understand directly.

In this table we have a relation of all the features available in MIR toolbox together with a description. We well mark in color the features picked for our classifier:

Descriptor name	**Description**
ENERGY	
mirrms	Root Mean Square, effective power of audio.
mirlowenergy	Number of frames with lower than average energy.
mirenvelope	Amplitude envelope (global shape of the waveform)
mironsets	Note onset positions and characteristics
mirattacktime	Duration of note attacks
mirattackslope	Average slope of note attacks
mirdecreaseslope	Note release phase description
mirattackleap	Change of amplitude in note attacks
mirduration	Note duration from attack to release
mireventdensity	Average frequency of events
RHYTHM	
mirtempo	Tempo (in beats per minute)
mirmetre	Metrical analysis
mirmetroid	Dynamic metrical centroid and strength
mirfluctuation	Fluctuation strength (periodicities in each channel)
mirbeatspectrum	Beat spectrum, characterizing the rhythmic content
mirpulseclarity	Rhythmic clarity, i.e., beat strength
TIMBRE	
mircentroid	
mirspread	
mirbrightness	Spectral brightness (high-frequency rate)
mirrolloff	Spectral rolloff (frequency above which is located a certain amount of energy)
mirmfcc	Mel-frequency cepstrum coefficients
mirinharmonicity	Inharmonicity (partials non-multiple of fundamental)
mirroughness	Roughness (sensory dissonance)
mirregularity	Spectrum irregularity (amplitude variability of successive peaks)
PITCH	
mirpitch	Pitch frequencies
mircepstrum	Cepstrum representation (showing periodicities)
mirmidi	Attempts a conversion of audio into MIDI
TONALITY	
mirchromagram	Chromagram (distribution of energy along pitches)
mirkeystrength	Key strengths (probability of key candidates)
mirkey	Best keys and modes (in the 12 tone system)
mirkeysom	Visualizes key strengths with self-organizing map
mirmode	General estimation of mode (major/minor)
mirtonalcentroid	Tonal centroid (using circles of fifths and thirds)
mirhcdf	Harmonic Change Detection Function

Table 3. MIR toolbox available descriptors and picked ones.

We said before that mood is spread and present along many features, we will now try to justify our feature selection for each category:

- Energy: Happy songs usually feel more en energetic, and will probably have a higher amount of instruments (more energy), but nowadays with the growing presence of Compression in music it's difficult to say if any differences in this facet could be found.
  Anyway, we are going to make use of some of them: RMS, LowEnergy, AttackTime, NoteDuration and EventDensity

- Timbre: Features based on Spectrum seem to be more representative of the "color" of the signal, and somehow model the shape of every sample so we are going to make use of all of them.

- Rhythm: Again, this kind of descriptors could be representative in our classification as we think that most of th Sad songs tend to be slower than happy ones, so we are going to use some related features as: Tempo,Metroid, and PulseClarity.

- Tonality: We don't know if key has a direct incidence on mood, but the mode is for sure something relevant in that case, as minor mode usually sounds more sad than major. We are going to consider: Key, Mode and TonalCentroid

As we commented in the beginning of this section, all the selected features are extracted with MATLAB (code attached in the .zip file), for all samples of each class. This process resulted in two .csv files (one for each class) containing all feature values (43 attributes) for each sample in the class (60 instances) [*Happy.csv* | *Sad.csv]*.

We then joined both .csv files and filtered some conflictive characters with LibreOffice Calc in order to get a file ready to be loaded in Weka [Features.csv].

The same procedure is done using the test dataset samples to get a Test .csv [Test.csv].

5. AUTOMATIC CLASSIFICATION

At this step, we have to model the extracted features of our training dataset together with their annotated class to build a classifier. To do it we used Weka.

After loading our training dataset, we applied a Normalize filter as suggested and performed Classification using different classifiers. We tried to used those which sound more common to us: J48 (decision tree), SMO (support vector machine), Ibk (k-Nearest Neighbours), AdaBoost and ZeroR as our Baseline.

The final classifier is evaluated both using Cross-validation (Xval) technique (10 folds) and with our Test Dataset, because an arbitrary and small test set was in our opinion not enough to evaluate the quality of a classifier.

Classifier	**Xval accuracy**	**Test D accuracy**
ZeroR (baseline)	*56,67 %*	*50 %*
J48	66,67 %	70 %
SMO	65 %	80 %
LBk	51,67%	40 %
AdaBoostM1	70 %	70 %

Table 4. Classifier accuracies in both evaluation methods.

Classifier	**Xval Confusion M**	**Test D Confusion M**
ZeroR (baseline)	a b <-- classified as 34 0 | a = Happy 26 0 | b = Sad	a b <-- classified as 5 0 | a = Happy 5 0 | b = Sad
J48	a b <-- classified as 24 10 | a = Happy 10 16 | b = Sad	a b <-- classified as 4 1 | a = Happy 3 2 | b = Sad
SMO	a b <-- classified as 25 9 | a = Happy 12 14 | b = Sad	a b <-- classified as 5 0 | a = Happy 3 2 | b = Sad
LBk	a b <-- classified as 26 8 | a = Happy 21 5 | b = Sad	a b <-- classified as 3 2 | a = Happy 4 1 | b = Sad
AdaBoostM1	a b <-- classified as 27 7 | a = Happy 11 15 | b = Sad	a b <-- classified as 4 1 | a = Happy 2 3 | b = Sad

Table 5. Confusion matrices in both evaluation methods.

ZeroR classifier just looks for the most popular class and guesses that all the time. It used a baseline to compare and evaluate the results for other classifiers. It can be easily understood by looking at the confusion matrix: all our instances get assigned to the most common class: Happy.

From the results we can say that both evaluation methods seem to be slightly correlated, as they tend to behave similarly along all classifiers.
The best classifier for Xval evaluation is AdaBoostM1which gets a 70% accuracy while SMO reaches 80% for our training dataset.

We are going to focus our analysis into the results of the Test dataset, as is a reduced, fixed and know bunch of samples that we can listen and analyze with more precision.
If we take a look to the accuracy by class of SMO and into it's confusion matrix, we can see that our mean results are around 80% both in Precision and Recall, having classified well all 'Happy' instances (recall = 1).

```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
                  1         0.4       0.714       1        0.833       0.8       Happy
                  0.6       0         1           0.6      0.75        0.8       Sad
Weighted Avg.     0.8       0.2       0.857       0.8      0.792       0.8
```

Figure 1. Caption of Weka Output: Datailed accuracy by class of SMO.

We could consider this results quite good, as our classifier will give 2 out of 10 erroneous results, but looking at cross validation or other classifier results, we can say that our features are not good enough to model this two classes, as overall results are not too high (around 60-70%).

As we used a small test dataset, we can now take a look to the specific result of each classified instance (using best-result methods):

Song	Annotated	Predicted
Soc feliç	Happy	Happy
Waterloo	Happy	Happy
Warmness on the Soul	Sad	Happy
Everybody Talk	Happy	Happy
My Immortal	Sad	Sad
Abril 74	Sad	Sad
Happy	Happy	Happy
Love of my life	Sad	Happy
Stay with me	Sad	Sad
Wake me up before you go go	Happy	Happy

Table 6. Annotated vs Predicted class with SMO classifier for each instance of the Test Dataset.

If we then go to Select Attributes, we will get a selection of those which perform better in our classification. In our case we get:

```
Selected attributes: 10,19 : 2
                     EventdensityMean
                     BrightnessMean
```

Figure 2. Caption of the Weka output: BestFirst attribute selection.

If we visualize them together in Weka:

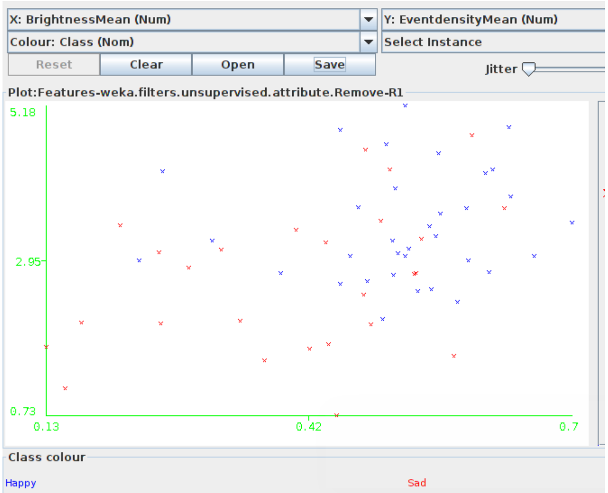

Figure 3. Weka plot caption: EventDensityMean vs BrightnessMean

We can see that it is a good starting point as there seems to be a slight tendency for Happy Songs to be Brighter (more high-frequency content) and with a higher Event Density Rate.

We can go a step further and perform a fast classification using only this two features [*Features_reduced.arff* | *Test_reduced.arff*]:

Classifier	Xval accuracy	Test D accuracy
ZeroR (baseline)	*56,67 %*	*50 %*
J48	73,33 %	60 %
SMO	71.67 %	70 %
LBk	58,33 %	50 %
AdaBoostM1	66,67 %	90 %

Table 7. Classifier accuracies in both evaluation methods for a reduced feature set.

If we compare with the previous result, we can easily see that despite we are using a drastically fewer number of features the results are not worse. This means that from all the selected features, this are actually the best pair. Additionally, we see that in some cases the new result is even better than the previous one, this can be due to the effect of over-fitting in the first classification where we had 43 attributes for 60 instances.

In this new scenario J48 is the best for Xval evaluation and AdaBoost for Test Dataset, where only one instance falls into error:

Song	Annotated	Predicted	Percentage
Soc feliç	Happy	Happy	0.667
Waterloo	Happy	Happy	0.853
Warmness on the Soul	Sad	Sad	0.598
Everybody Talk	Happy	Happy	0.853
My Immortal	Sad	Happy	0.667
Abril 74	Sad	Sad	0.598
Happy	Happy	Happy	0.667
Love of my life	Sad	Sad	0.988
Stay with me	Sad	Sad	0.598
Wake me up before you go go	Happy	Happy	0.667

Table 8. Annotated vs Predicted class with SMO classifier for each instance of the Test Dataset for a reduced feature set.

6. CONCLUSIONS

We started the whole process with the idea of getting an insight on automatic classification. We worked on mood, a very subjective taxonomy which apart from the content, have a lot of information in Context (both user and music).

We picked a huge amount of features that could be somehow related to the selected classes Happy and Sad, but working with a small Training Dataset of 60 instances, our picked 43 features seem to over-fit some of the classifiers.

After testing different classifiers and analyzing results EventDensityMean and BrightnessMean appeared the most relevant features from our selection, leading a 2-feature classifier to surprisingly good results compared to the complete feature selection.