I have decided to work in the content analysis for genre classification.

Given that dataset in which we havve a genre ground truth I try to study the diferent descriptors that allow us to classify the songs by genre.

The main difficult in that work is that non-linear SVMs are difficult to analyze, so we don’t know which descriptors contribute to the separation between classes (genres).

To analyze this big dataset, I need to organize correctly it and be able to extract the descriptors that I want to study. In order to do this work I create a python code (attached in a gib repository).

The first step was read the original csv file, where we have a dataset of 6300 songs (450 songs for each genre = asian, blues, classical, hip hop, rnb, rock, pop, easy-listening, folk, ska, jazz, electronic, country and clatinamer) and split it by items

with open("last\_fm.csv", "rb") as f:

line=[line.split() for line in f]

CorList = []

for i in range (0,6301):

CorLine = []

CorLine.append(str(line[i]).split(','))

CorList.append(str(CorLine[0]).split(','))

Then I choose six categories: classical, electronic, hip hop, rnb, jazz and folk. They are really different so it will be a good point to start to understand what descriptors works better to classify the different classes. And instead working with all the six categories, I compare it two by two or three by three.

In this point, I create six different matrix to save all the descriptors for each class.

classicalList = []

for i in range(901,1351):

for j in range(2,270):

classicalList.append(CorList[i][j][2:len(CorList[i][j])-1])

And the last work was assign to a new vector (classicaldef) only few of the descriptors

classicaldef[i][0] = 1

classicaldef[i][1] = classical[i][14]

classicaldef[i][2] = classical[i][47]

classicaldef[i][3] = classical[i][63]

classicaldef[i][4] = classical[i][70]

classicaldef[i][5] = classical[i][84]

classicaldef[i][6] = classical[i][119]

classicaldef[i][7] = classical[i][98]

classicaldef[i][8] = classical[i][189]

With this numbers I can choose the descriptors that I want to analyze. SVM only uses the mean and variance values of each feature, so you should only consider these descriptors.

Finally I extract the new csv to put in weka

with open("groundtruth.csv", "wb") as csvfile:

fieldnames = ['class', 'average\_loudness', 'spectral\_spread\_var', 'ZCR\_mean',...]

writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

writer.writeheader()

writer.writerow({'class': 'classical', 'average\_loudness': classicaldef[i][1],…]

Descriptors

After several test, I think that these descriptors are the most significants in order to separate the songs by genre:

average\_loudness

spectral\_spread\_var

ZCR\_mean

Dissonance\_mean

spectral\_flux\_mean

spectral\_centroid\_mean

energyband\_middle\_high\_mean

energyband\_middle\_low\_mean

Results:

Working with three classes I have a 76.44% of accuracy. We can see the confussion matrix below

348 23 79 | a = classical

6 374 70 | b = hiphop

39 101 310 | c = rnb

The most error was classifaying rnb as hip hop, what is a comprehensive error.

Working with four classes I have a 62,5% of accuracy. We can see the confussion matrix below

310 23 44 73 | a = classical

5 364 61 20 | b = hiphop

27 95 243 85 | c = rnb

97 39 106 208 | d = folk

Working with three classes I have a 51.15% of accuracy. We can see the confussion matrix below

309 24 39 20 58 | a = classical

3 360 57 18 12 | b = hiphop

27 94 237 36 56 | c = rnb

124 37 115 87 87 | d = jazz

97 35 96 64 158 | e = folk

Working with three classes I have a 43.29% of accuracy. We can see the confussion matrix below

308 14 14 38 19 57 | a = classical

36 44 237 75 18 40 | b = electronic

3 27 352 43 14 11 | c = hiphop

27 23 83 231 33 53 | d = rnb

124 11 35 112 81 87 | e = jazz

96 16 26 100 59 153 | f = folk

The algorithm that perfoms better is sequential minimal optimization