Information Retrieval

Assignment 3, summer semester 2011

by Patrick Marschik (0625039), Benjamin Bachhuber (1028430)

Table of Contents

Similarity Retrieval

Task Merged Result List

A merged result list; if you have e.g. three feature sets, and you retrieve k documents from each, you most likely will end up with three different results, and a total number of >= k documents. You shall think of a way to unify the information from all three lists into one final result list; we don't demand a specific solution, but you shall argue in your report why you chose your solution.
For a selected set of 10 (randomly chosen) queries, compare the results of the merged result list with those of the single feature sets. Include and describe this comparison in the report.

At first a result list with length k is generated for every feature.

Sample Result:

```
query: SlowWaltz/SlowWaltzSlowWaltz_Fire13.mp3
rank ISMI.rh.arff                                 ISMI.rp.arff
-----+----------------------------------------+---------------------------------------
#  0 SlowWaltz/SlowWaltzSlowWaltz_1   1,813492 SlowWaltz/SlowWaltzSlowWaltz_B  85,419958
#  1 SlowWaltz/SlowWaltzSlowWaltz_B   1,846857 Rumba/RumbaRumba_Pais_Tropical  86,168813
#  2 SlowWaltz/SlowWaltzSlowWaltz_1   1,864096 SlowWaltz/SlowWaltzSlowWaltz_M  87,223598
#  3 SlowWaltz/SlowWaltzSlowWaltz_M   1,953502 SlowWaltz/SlowWaltzSlowWaltz_C  87,267336
#  4 SlowWaltz/SlowWaltzSlowWaltz_1   2,039259 SlowWaltz/SlowWaltzSlowWaltz_1  87,318169
#  5 SlowWaltz/SlowWaltzSlowWaltz_B   2,051620 VienneseWaltz/VienneseWaltzVie  87,373909
#  6 SlowWaltz/SlowWaltzSlowWaltz_1   2,104884 SlowWaltz/SlowWaltzSlowWaltz_1  87,594618
```

```
ISMI.ssd.arff
+----------------------------------------+
Rumba/RumbaRumba_Fire03.mp3     14,067746
Rumba/RumbaRumba_Latin_Jam307.  16,869131
Rumba/RumbaRumba_Fire-10.mp3    17,106871
SlowWaltz/SlowWaltzSlowWaltz_B  17,287559
SlowWaltz/SlowWaltzSlowWaltz_1  17,310635
SlowWaltz/SlowWaltzSlowWaltz_1  18,170017
SlowWaltz/SlowWaltzSlowWaltz_1  18,508561
```

It is important to unify the results of all the feature lists into one merged result list. To make this possible it’s necessary to normalize the distances of the results in the different result lists.
A min-max normalization was used with a desired output range 0 to 1. To do this it’s necessary to read out the minimal and maximal distances of each of the feature lists:

```
Distances:
Index: features\ISMI_output\ISMI.rh.arff Min: 1.8134924626365876 Max:
17.66140958442694 Avg: 6.7834110362047975
Index: features\ISMI_output\ISMI.ssd.arff Min: 14.0677461935763 Max:
55.20229701054049 Avg: 31.84543669939567
Index: features\ISMI_output\ISMI.rp.arff Min: 85.41995791974807 Max:
317.6972700455886 Avg: 148.23716317420482
Average -  Min: 33.76706552532032 Max: 130.18699221351866 Avg: 62.288670303268425
```

Afterwards it’s possible to rank the results in one merged result list using the normalized distances. A result document can appear more than one time in the merged list. So it is important to remove duplicates. This is done by removing the duplicate with the higher distance. For the example above a result might look like this:

```
Merged Result List:
1 SlowWaltz/SlowWaltzSlowWaltz_105502.mp3                                 0,000000
2 SlowWaltz/SlowWaltzSlowWaltz_Ballroom_Classics402.mp3                   0,000000
3 Rumba/RumbaRumba_Fire03.mp3                                             0,000000
4 SlowWaltz/SlowWaltzSlowWaltz_Ballroom_Magic02.mp3                       0,002105
5 SlowWaltz/SlowWaltzSlowWaltz_105106.mp3                                 0,003193
6 Rumba/RumbaRumba_Pais_Tropical-12.mp3                                   0,003224
7 SlowWaltz/SlowWaltzSlowWaltz_Media-103303.mp3                           0,007765
```

Task - Additional analysis over all query documents

- *the min/max/avg distances averaged over all documents in a specific set*

ISMI Dataset:

```
Distances per class:
Class: Tango Min: 0.0 Max: 1.0 Avg: 0.19207294353081222
Class: Rumba Min: 0.0 Max: 0.9572091013819751 Avg: 0.22455538017449334
Class: Jive Min: 0.0 Max: 1.0 Avg: 0.2783164992452297
Class: SlowWaltz Min: 0.0 Max: 1.0 Avg: 0.14918449315649088
Class: VienneseWaltz Min: 0.0 Max: 0.8000674667416143 Avg: 0.170470244861017
Class: Samba Min: 0.0 Max: 0.9095484448766152 Avg: 0.24437336575564883
Class: Quickstep Min: 0.0 Max: 0.944130603992192 Avg: 0.21471300751982295
Class: ChaChaCha Min: 0.0 Max: 1.0 Avg: 0.2852635406412479
```

The SlowWaltz genre has the lowest average distance. Therefore the songs of this class are most similar.The ChaChaCha class on the other hand seems to contain a wide variety of different songs.

GTZAN Dataset:

```
Distances per class:
Class: reggae Min: 0.0 Max: 1.0 Avg: 0.2874583769502494
Class: classical Min: 0.0 Max: 1.0 Avg: 0.1517387872375771
Class: hiphop Min: 0.0 Max: 1.0 Avg: 0.27945827626683484
Class: metal Min: 0.0 Max: 0.7456662929173176 Avg: 0.17594781820492109
Class: jazz Min: 0.0 Max: 0.8265824978051459 Avg: 0.20827035288450954
Class: disco Min: 0.0 Max: 0.7319065069579245 Avg: 0.2338231194632184
Class: pop Min: 0.0 Max: 0.8663555199227382 Avg: 0.2771621867649723
Class: rock Min: 0.0 Max: 0.6609374239646594 Avg: 0.16795978923974164
Class: blues Min: 0.0 Max: 0.8769981135258997 Avg: 0.21223549249462836
Class: country Min: 0.0 Max: 0.7811432035215174 Avg: 0.15393146097747293
```

Since the classical music genre has the lowest average distance it seems that the classical songs of the test dataset are very similar to each other.
The hip-hop genre has the highest average distance which can be explained by the diversity among the songs of this genre.

- *how often a document ocurrs in all the result lists provided by all other documents.*
- *I.e. if you have n documents, and you don't show a document in the result list where you use that document as a query, then there are n-1 queries that could potentially retrieve a document.*
- *Record how many times l <= (n-1) a document appears. Compute this statistic for each of the result lists (i.e. each feature set, and also the merged result list if you cut that one of at k documents as well).*

The retrieval outputs also contain information about the occurrences of each document.

One list shows the occurrence of a document in all merged result lists, e.g.:

```
Occurrences of documents in other result lists:
Document: Quickstep/QuickstepQuickstep_104917.mp3 - Occurrences: 1
Document: SlowWaltz/SlowWaltzSlowWaltz_Commitments-11.mp3 - Occurrences: 1
Document: ChaChaCha/ChaChaChaChaChaCha_106117.mp3 - Occurrences: 1
Document: ChaChaCha/ChaChaChaChaChaCha_Latin_Jam07.mp3 - Occurrences: 1
Document: Rumba/RumbaRumba_Latino_Latino-05.mp3 - Occurrences: 1
Document: Tango/TangoTango_Ballroom_Magic08.mp3 - Occurrences: 1
Document: Quickstep/QuickstepQuickstep_104218.mp3 - Occurrences: 1
Document: Samba/SambaSamba_103903.mp3 - Occurrences: 1
....
Document: SlowWaltz/SlowWaltzSlowWaltz_Ballroom_Magic01.mp3 - Occurrences: 58
Document: Tango/TangoTango_Chrisanne106.mp3 - Occurrences: 61
Document: Tango/TangoTango_StrictlyDancing_Tango10.mp3 - Occurrences: 62
Document: SlowWaltz/SlowWaltzSlowWaltz_Chrisanne101.mp3 - Occurrences: 63
Document: SlowWaltz/SlowWaltzSlowWaltz_104302.mp3 - Occurrences: 63
Document: Quickstep/QuickstepQuickstep_Chrisanne211.mp3 - Occurrences: 78
```

Others show the occurrence in each of the feature based result lists, e.g.:

```
Occurrence of Documents in feature features\ISMI_output\ISMI.rh.arff:
Document: ChaChaCha/ChaChaChaChaChaCha_Media-103617.mp3 - Occurrences: 1
Document: Quickstep/QuickstepQuickstep_105516.mp3 - Occurrences: 1
Document: ChaChaCha/ChaChaChaChaChaCha_Cafe_Paradiso-06.mp3 - Occurrences: 1
Document: Rumba/RumbaRumba_106119.mp3 - Occurrences: 1
Document: ChaChaCha/ChaChaChaChaChaCha_Media-106105.mp3 - Occurrences: 1
....
Document: Tango/TangoTango_Media-104707.mp3 - Occurrences: 26
Document: ChaChaCha/ChaChaChaChaChaCha_Pais_Tropical05.mp3 - Occurrences: 27
Document: Rumba/RumbaRumba_Pais_Tropical11.mp3 - Occurrences: 28
Document: Jive/JiveJive_Media-106115.mp3 - Occurrences: 28
Document: Quickstep/QuickstepQuickstep_Media-104418.mp3 - Occurrences: 32
```

We see that for RH and SSD the number of occurrences in the top of the list are distributed relatively even whereas for the RP feature-set the occurrences of about 5 documents stand out (e.g. for GZTAN they have 96, 2 time 7x and 2 times 6x while the rest is about 40-50). So RP often returns the same document which could mean that these documents are on the cluster boundaries of its own cluster and some other clusters.

Comparison of the different features

```
query: metal/metalmetal.00008.mp3
rank GTZAN.rh.arff                                GTZAN.rp.arff
-----+-----------------------------------------+----------------------------------------
#  0 metal/metalmetal.00044.mp3       1,655032 rock/'rock\rock.00059.mp3'      74,539509
#  1 metal/metalmetal.00065.mp3       1,655032 pop/poppop.00014.mp3            74,812653
#  2 metal/metalmetal.00070.mp3       2,091486 rock/'rock\rock.00096.mp3'      75,975025
#  3 metal/metalmetal.00039.mp3       2,108954 metal/metalmetal.00006.mp3      76,966466
#  4 metal/metalmetal.00041.mp3       2,151252 metal/metalmetal.00044.mp3      78,634093
#  5 metal/metalmetal.00062.mp3       2,151252 metal/metalmetal.00065.mp3      78,634093
#  6 metal/metalmetal.00006.mp3       2,178349 metal/metalmetal.00001.mp3      79,216835
```

```
GTZAN.ssd.arff
+-----------------------------------------+
metal/metalmetal.00004.mp3      11,535612
metal/metalmetal.00006.mp3      11,564422
jazz/jazzjazz.00079.mp3         12,085950
jazz/jazzjazz.00084.mp3         12,795801
jazz/jazzjazz.00082.mp3         12,815440
metal/metalmetal.00074.mp3      13,358275
metal/metalmetal.00007.mp3      13,501384
```

Most of the retrieved result lists show that a retrieval based on the rh feature returns songs from the same genre in most of the cases. The retrievals based on the rp feature perform almost as well as the rh feature. Results based on the ssd feature show the biggest diversity since a lot of songs from different genres can be found among the top ranked results.

```
Distances:
Index: features\GTZAN_output\GTZAN.rp.arff Min: 74.5395092239484 Max: 279.75411264768644
Avg: 128.5452665672904
Index: features\GTZAN_output\GTZAN.ssd.arff Min: 11.535612480249906
Max: 61.396000377536815 Avg: 23.678720408315844
Index: features\GTZAN_output\GTZAN.rh.arff Min: 1.6550315028380624
Max: 14.057164869132743 Avg: 5.7465054143935355
Average -  Min: 29.243384402345455 Max: 118.40242596478534 Avg: 52.656830796666604
```

```
Merged Result List:
1 metal/metalmetal.00044.mp3                                                 0,000000
2 metal/metalmetal.00065.mp3                                                 0,000000
3 rock/'rock\rock.00059.mp3'                                                 0,000000
4 metal/metalmetal.00004.mp3                                                 0,000000
5 metal/metalmetal.00006.mp3                                                 0,000578
6 pop/poppop.00014.mp3                                                       0,001331
7 rock/'rock\rock.00096.mp3'                                                 0,006995
```

Generating a merged result list by using min-max normalization of the distances seems to deliver good results. In the example above, just results from the same genre or a similar genre are returned.
Other merged result lists deliver a similar good result. It seems that even though the results based on the ssd feature show some songs from totally different genres this does not have an effect on the results in the merged result list.

Genre Classification

k-Nearest Neighbors

From our results we can see that changing the algorithm to search neighbors (linear NN vs. KD-Tree) does not change the resulting percentage of correct results. Changing the number of neighbors used for classification however changes the results: less neighbors (1 vs. 3) generally decrease the correctness which is due to the fact that items on cluster borders will probably get misclassified.

Support Vector Machines

For SVMs we can see that a normalized polynomial kernel is worse than a non-normalized one. This is because the default settings (as requested by the instructions) for the normalized polynomial kernel the exponent is set to 2 and for non-normalized kernels it is 1. It therefore seems that a 1st order polynomial is better suited to music classification then a 2nd order polynomial.
We observed that a higher value for the SVM’s complexity (5 vs. 1) yields worse results. This is only natural as C controls the trade-off between the margin and the error penalty and a higher value allows for more errors.

Random Forests

Increasing the number of features (0 vs. 2 where 0 corresponds to unlimited) improved the results as 2 features seem too few to classify the samples which contain of 60-1380 features.
Increasing the number of trees from 10 to 30 - as expected - boosted the number of correctly classified samples.

Method Comparison

Overall we can see that the SVMs produce the best results, followed by kNN and the worst results are delivered by Random Forests. The bad results of Random Forests can be explained because they only separate the features in a linear manner which doesn’t seem to lend itself to music classification.

Feature Comparison

For the ISMI dataset the features extracted by RH produce the best results for all algorithms. Interestingly for GTZAN SSD outperforms RH and RP for all algorithms. This might be because ISMI is a quite homogenic dataset (only containing ballroom-dance music) wheras GTZAN is more heterogen. Therefore it seems SSD offers itself to be used for diverse musical data wheras RH might be useful in settings where similar music needs to be classified. RP delivers similar results to RH in both datasets.