|  |
| --- |
| Fakultet elektrotehnike i računarstva – strojno učenje |
| Music Genre Classifier |
| Dokumentacija |

|  |
| --- |
| Đuretec, Fabek, Gulić, Lučanin, Vidačić  2009. |

Sadržaj

[Teorija 3](#_Toc249340182)

[Arhitektura sustava 4](#_Toc249340183)

[Sonic Annotator 4](#_Toc249340184)

[ID3/kazalo izlučivač žanrova 4](#_Toc249340185)

[Korišteni alati 5](#_Toc249340186)

[Sonic Annotator 5](#_Toc249340187)

[Vamp plugins 5](#_Toc249340188)

[Korištene značajke 6](#_Toc249340189)

[Upute za instalaciju i korištenje 7](#_Toc249340190)

[Instalacija Vamp priključaka 7](#_Toc249340191)

[Pokretanje aplikacije 7](#_Toc249340192)

[Konfiguracijska datoteka 7](#_Toc249340193)

[Obrazac korištenja aplikacije 1 - testiranje klasifikacije 10](#_Toc249340194)

[Obrazac korištenja aplikacije 2 - ponovno testiranje s drugim parametrima - ubrzanje 11](#_Toc249340195)

[Rezultati 12](#_Toc249340196)

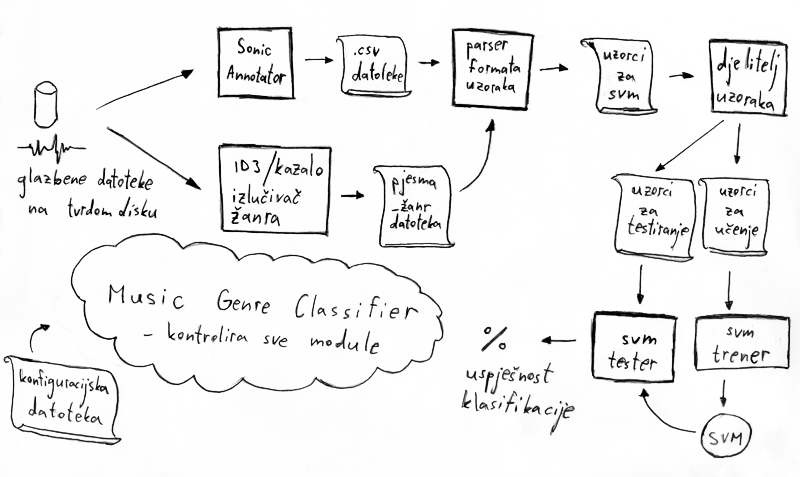
## Teorija

## Arhitektura sustava

U našem projektu aplikaciju smo razvijali tako da smo razbili funkcionalnost na više manjih modula koji obavljaju točno određene zadatke.

Sama izvršna datoteka Music Genre Classifier je tako u stvari samo jedna skripta koja kao ulaz prima konfiguracijsku datoteku "conf-file.txt" i kontrolira sve druge module (poziva ih).

Arhitektura ovog sustava se najbolje može vidjeti iz grafičkog prikaza.



Slika - slika arhitekture našeg sustava

U sljedećim cjelinama se ukratko s visoke razine objašnjavaju detalji pojedinih modula.

### Sonic Annotator

Jedini modul koji nismo sami implementirali, već smo koristili gotov produkt. Detalji korištenja Sonic Annotatora nalaze se u zasebnom poglavlju. Ukratko - kao izlaz daje tekstualnu datoteku za svaku obrađenu pjesmu s izračunatim vrijednostima značajki.

### ID3/kazalo izlučivač žanrova

Dio je same skripte prevedene u izvršnu datoteku „Music Genre Classifier.exe“. Rekurzivno čita sve datoteke iz ulaznog glazbenog kazala i određuje "pravi" žanr svake pjesme (koja je jednog od podržanih formata - zadano u „conf-file.txt“) na jedan od 2 načina:

* čitajući žanr (engl. genre) id3 tag glazbene datoteke
* gledajući naziv vršnog kazala (dakle pretpostavlja se da su sve pjesme rock žanra negdje u kazalu naziva „rock“)

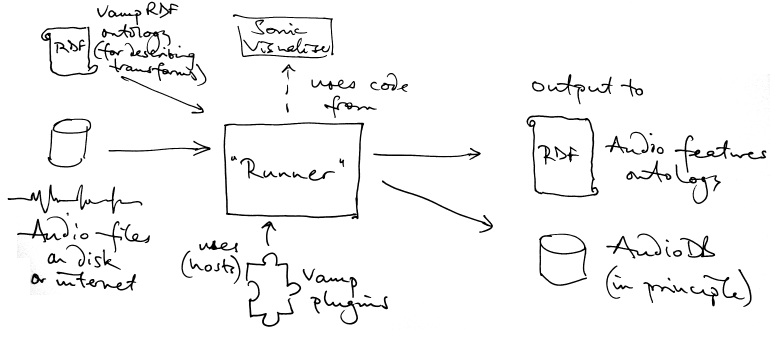
Kao izlaz daje datoteku „uzorak\_razred.txt“ u kojoj piše žanr svake pjesme. Ovi podaci će se kasnije koristiti kao ocjenjivač pri treniranju i testiranju klasifikatora.

## Korišteni alati

Sonic Annotator i Vamp plugins - izlučivanje značajki

### Sonic Annotator

Sonic Annotator (ranije znan kao „Runner“) je program naredbenog retka (engl. command line) za izlučivanje glazbenih značajka iz više glazbenih datoteka. Osnovna ideja je pružiti proces izvlačenja značajki, a pritom sakriti konkretne metode, koristeći Vamp priključke (engl. Vamp plugins) za izlučivanje značajki. Izlazni formati su RDF, CSV (zarezom odvojene vrijednosti, engl. comma separated values).



Slika - arhitektura Sonic Annotatora

Sonic anotator kao ulaz koristi:

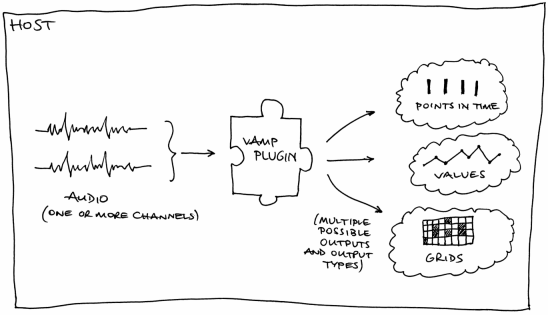
- RDF datoteku (tzv. kostur, engl. skeleton) koje opisuju Vamp priključke na koje se treba spojiti.

- lokaciju glazbenih datoteka koje treba obraditi (mogu biti čak i na Internetu)

Izlaz je skup .csv datoteka (ili drugačijih, ovisi kako se zada) za svaku pjesmu i svaku značajku koje sadrže vrijednosti tih (što te vrijednosti točno znače, zavisi od korištenih značajki).

### Vamp plugins

Vamp je sustav priključaka za obradu zvuka koji izlučuje opisne informacije iz zvučnih podataka - drugim riječima vrši zvučnu analizu ili izlučivanje značajki.



Slika - arhitektura Vamp priključaka

U našem projektu nisu bili direktno korišteni, već kroz Sonic Annotator, na koji se može gledati kao sučelje prema ovim priključcima.

Priključci koje smo instalirali za potrebe projekta su „Vamp libxtract plugins“.

To su priključci za izlučivanje značajki niske razine koji koriste libxtract biblioteku Jamiea Bullocka kako bi pružili oko 50 spektralnih i drugih značajki.

### Korištene značajke

Pri testiranju smo kao značajke koristili mfcc (engl. Mel-frequency cepstral coefficients), koje se i u literaturi preporučaju za klasifikaciju glazbe. Računali smo ih na 3 intervala trajanja 5 sekundi u različitim dijelovima pjesme (<0,5> sec, <20,25> sec i <50,55> sec).

Dobivanje značajki mfccom je pobliže objašnjeno u teorijskom dijelu dokumentacije.

Sonic Annotator smo koristili naredbom:

*sonic-annotator.exe -t mfcc -S mean --summary-only --segments 5,20,25,50,55 -w csv --csv-force --csv-basedir . -r d:\staza\do\kazala*

Datoteka „mfcc“ je kostur za istoimeni Vamp priključak, a dobiva se izvođenjem naredbe:

*sonic-annotator -s vamp:vamp-libxtract:mfcc:mfcc > mfcc*

Oznake „mean“, „summary-only“ i „segments“ služe da se smanji inače jako velik broj vrijednosti. Segments određuje granice za vremenske intervale u kojima se onda svake sekunde radi sažetak (engl. summary) tako da se izračuna aritmetička sredina (engl. mean) svih vrijednosti.

U projektu smo kasnije parsirali izlazne .csv datoteke i uzeli samo vrijednosti iz 3 prije spomenuta intervala (zanemarujući ostale intervale, npr. <5,20> sec).

Tim procesom za svaku glazbenu datoteku (pod uvjetom da je duža od 55 sekundi) dobijemo 63 značajke. Za SVM klasifikator se i inače preporuča stotinjak vrijednosti po uzorku.

## Upute za instalaciju i korištenje

Kako aplikacija zavisi o programu Sonic Annotator za izvlačenje značajki iz glazbenih datoteka, a Sonic Annotator nadalje koristi Vamp priključke (engl. Vamp plugins), potrebno je imati Vamp priključke na točno zadanom mjestu. Sam Sonic Annotator ne treba posebno instalirati jer je to samo jedna izvršna datoteka koja se nalazi u „working bundle“ kazalu.

### Instalacija Vamp priključaka

1. skinite željene Vamp priključke (u projektu smo koristili libxtract) sa ove stranice <http://www.vamp-plugins.org/download.html>
2. raspakirajte .rar datoteku i stavite.dll i .cat datoteke u kazalo „C:\Program Files\Vamp Plugins“ - ako koristite 64-bitni Windows OS, umjesto prethodnog koristite kazalo "C:\Program Files (x86)\Vamp Plugins"
3. testirajte dostupnost Vamp priključaka tako što se u naredbenom retku (engl. command prompt) pozicionirate na „working bundle“ kazalo našeg projekta (gdje je i sonic-annotator.exe) i pokrenete Sonic Annotator naredbom: „sonic-annotator.exe –l (ukoliko se ne ispisuju razni vamp:libxtract priključci (osim „example“ priključaka) onda nešto nije prošlo dobro u koraku 2)

### Pokretanje aplikacije

Sve izvršne datoteke (i jedna konfiguracijska) potrebne za pokretanje aplikacije (Sonic Annotator također) dostupne su u kazalu „working bundle“.

Glavna skripta koja kontrolira sve druge dijelove programa je prevedena u izvršnu datoteku „music\_genre\_classifier.exe“. Potpunu funkcionalnost programa moguće je vidjeti njenim pokretanjem (najbolje u naredbenom retku za bolji pregled ispisa).

### Konfiguracijska datoteka

Sve postavke i parametre aplikacije podešava se kroz tekstualnu datoteku „conf-file.txt“. U komentarima datoteke objašnjeno je ukratko što točno koji parametar znači i koje su moguće vrijednosti.

NAPOMENA: poredak redaka u konfiguracijskoj datoteci je bitan i ne smije ga se mijenjati!

Primjer izgleda konfiguracijske datoteke:

##

## KONFIGURACIJSKA DATOTEKA za skriptu koja kontrolira pojedine dijelove aplikacije

##

# direktorij gdje se nalazi glazba za učenje i testiranje

D:\glazba

# tipovi glazbe koji se žele podržati (ostali se ignoriraju) - sonic annotator isto mora podržavati tip

.mp3 .wav

# žanrovi koji se žele podržavati

classical,rock,jazz

# naziv datoteke u koju će se zapisati pripadnost datoteke razredu (NE MIJENJATI)

uzorak\_razred.txt

# željena metoda za određivanje žanra: folder ili id3

# ili na eng...

# desired method for determining the actual genre (for validation purposes): "folder" or "id3"

# with "folder" all the songs are supposed to be in a folder of it's genre's name

# with "id3" the value is extracted from id3 tags

folder

# pozovi sonic annotator (ovisno o broju datoteka može trajati jako dugo): 1 ili 0

# ako se zada 0, potrebno je ručno kopirati neke unaprijed izračunate značajke

1

# naredba koja se koristi za sonic annotator (bez staze do foldera s muzikom, to se konkatenira na kraj)

sonic-annotator.exe -t mfcc -S mean --summary-only --segments 5,20,25,50,55 -w csv --csv-force --csv-basedir . -r

# naredba za skaliranje

svm-scale.exe -l 0 -u 1

# klasifikator: libsvm ili kreshvm

# "libsvm" znači koristi biblioteku libsvm

# "kreshvm" je naš klasifikator

kreshvm

# naredba za libsvm-ov svm-train

svm-train -s 0 -c 100 -g 0.1 -v 5

# naredba za treniranje našeg klasifikatora kreshvm

# kreshvm\_train.exe 63 0.5 1

# kreshvm\_train parametri

# drugi parametar - konstanta C

5

# drugi parametar - tip svm-a

# 1 - linearni

# 2 - rbf kernel

# 3 - polinomni kernel

1

# treći parametar - gama za rbf, stupanj za polinomni kernel, nebitno za linearni svm

4.3

# testiranje našeg klasifikatora

kreshvm\_test.exe

#naredba za odvajanje skupa podataka na učenje i testiranje

odvoji.exe 80%

### Obrazac korištenja aplikacije 1 - testiranje klasifikacije

Pretpostavimo da se na računalu pod "D:\glazba" nalaze 3 kazala „jazz“, „classical“ i „rock“ u kojima se dalje nalaze kazala izvođača i albuma (proizvoljne dubine, npr. „D:\glazba\rock\Led Zeppelin\The Song Remains The Same\Rain Song.mp3“ je dozvoljena staza), a negdje u njima razne glazbene datoteke odgovarajućeg žanra (dakle sve rock pjesme su negdje u folderu „rock“).

Mi želimo naučiti SVM s 80% tih glazbenih datoteka (ravnomjerno odabranih pripadnika svakog žanra) i testirati ga s preostalih 20%.

Tada postavljamo sljedeće parametre konfiguracijske datoteke conf-file.txt:

- ulazni glazbeni direktorij - D:\glazba

- žanrovi koji se žele podržavati - jazz,rock,classical

(moraju se poklapati s kazalima žanrova)

- željena metoda za određivanje žanra - folder

(zato što kazalo (engl. folder) određuje ispravni žanr, a ne id3 tag)

- naredba za odvajanje skupa podataka na učenje i testiranje - odvoji.exe 80%

- pozovi sonic annotator - 1

(ako nemamo od prije izračunate značajke za glazbene datoteke)

- klasifikator - kreshvm

(ovo je naš klasifikator pa vjerojatno njega želimo testirati)

Ostale vrijednosti su parametri drugim dijelovima aplikacije te su detaljnije objašnjene u drugim dijelovima dokumentacije, a pretpostavljene vrijednosti bi trebale biti dovoljno dobre u većini slučajeva.

Kako to točno zapisati u "conf-file.txt" vidljivo je u prethodnom odjeljku jer upravo ta datoteka odgovara ovom primjeru.

Nakon što smo to podesili dovoljno je pokrenuti "music\_genre\_classifier.exe" i ako sve odradi dobro, na kraju će se ispisati postotak uspješno klasificiranih testnih glazbenih datoteka (onih 20% koje nisu bile korištene za učenje SVM-a).

### Obrazac korištenja aplikacije 2 - ponovno testiranje s drugim parametrima - ubrzanje

Recimo da želimo promijeniti neki parametar i ponovno testirati klasifikator, s novim parametarom korištenim pri učenju. Daleko najsporiji dio rada aplikacije je izlučivanje značajki uz pomoć Sonic Annotatora.

Da ne bi ponovno morali čekati vrlo dug proces obrade glazbenih datoteka, možemo ubrzati proces tako da ne pozivamo Sonic Annotator.

To se postavlja u konfiguracijskoj datoteci pišući pod **pozovi sonic annotator - 0**

NAPOMENA: ukoliko se želi koristiti ovo ubrzanje, u kazalu "working bundle" zajedno s izvršnim datotekama moraju biti **.csv datoteke** svih onih glazbenih datoteka koje želimo koristiti za treniranje i testiranje i **uzorak\_razred.txt datoteka** koja sadržava sve te pjesme s oznakama žanrova. Ako ništa nije brisano od zadnjeg računanja, ovaj uvjet bi trebao biti zadovoljen.

Kako bi koristili već izračunate značajke (i ubrzali testiranje različitih parametara svm-a) pripremljena je zaliha unaprijed izračunatih značajki u kazalu „precomputed examples“ (najviše za GTZAN). Te značajke se mogu iskoristiti tako da se odabere neku od ovih baza (npr. gtzan) i iz kazala "gtzan" koji se nalazi u kazalu "precomputed examples" kopira u "working bundle" kazalo datoteku uzorak\_razred.txt i sve značajke koje se žele računati.

Ako bi npr. htjeli računati klasifikaciju po mfcc-u i nonzero-count-u iskopirali bi sadržaje ta 2 kazala u "working bundle".

## Rezultati

Točnost klasifikacije je računata tako da se skup uzoraka (pjesama) podijelio na skup za treniranje i skup za testiranje u nekom omjeru (koji će biti naveden, uglavnom je 80% korišteno za treniranje). Presjek ta dva skupa je uvijek prazan skup. Ta se metoda naziva još i hold-out metoda jer je jedan dio skupa ostao „tajan“ i korišten samo za testiranje.

Točnost predstavlja omjer točno klasificiranih uzoraka i svih uzoraka iz skupa za testiranje.

### Baze pjesama

Za testiranje smo koristili više baza:

1. ISMIR – Na konferenciji ISMIR 2004. je objavljena baza pjesama za natjecanje u raspoznavanju žanrova. Sastoji se od 729 pjesama u 8 žanrova. Mana je što neki razredi imaju svega nekoliko pjesama (pop npr. ima samo 6 pjesama). Baza je dostupna za skidanje na adresi <http://ismir2004.ismir.net/genre_contest/index.html>
2. GTZAN – Ovaj skup pjesama je bio korišten za rad u raspoznavanju žanrova „Musical genre classification of audio signals“ G. Tzanetakisa i P. Cooka. Dobro je što od svakog žanra ima po 100 isječaka pjesama (u trajanju od 30 sekundi), no mana je što su stavili dosta (čak i za ljude) sličnih žanrova – recimo blues,country, reggae, pop, disco. Razlike između nekih pjesama tih žanrova su prilično diskutabilne što se može vidjeti i iz rezultata klasifikacije. Može se skinuti na adresi <http://marsyas.sness.net/download/data_sets>
3. osobna – Ovaj skup smo sami sastavili od 57 pjesama, žanrova rock, classical i jazz (19 pjesama svaki žanr) i predstavlja malo lakši primjer.

Rezultati s variranjem baza:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| baza | broj žanrova | značajka | klasifikator | parametri s kojima je dobivena najbolja točnost | treniranje (%) | najbolja dobivena točnost klasifikacije (%) |
| ismir | 8 | mfcc | kreshvm | C=2, rbf, gama=0.1 | 80 | 56,7376 |
| gtzan | 10 | mfcc | kreshvm | C=0.6, polinomni, 1. Stupanj | 80 | 48 |
| osobna | 3 | mfcc | kreshvm | C=2, rbf, gama=0.1 i također C=0.1, linearni | 80 | 91,6667 |

Iz ispisa klasifikatora su se potvrdile slutnje navedene kao mane u opisima baza. Samo je za našu, „relativno čistu“ bazu klasifikator imao svega jednu grešku.

### Značajke

Izbor značajki je jedna od presudnih elemenata uspješne klasifikacije. Ovdje je dan pregled njihovih različitih kombinacija (kada je navedeno više značajki, znači da se za svaki uzorak izračunaju sve značajke i nanižu). Broj vrijednosti je ukupan broj vrijednosti svih korištenih značajki koje se dobiju za svaki uzorak.

Korištena je GTZAN baza, 80% treniranje, a parametri svm-a su: C=0.6, polinomni kernel 1. stupnja.

Rezultati s variranjem značajki:

|  |  |  |
| --- | --- | --- |
| značajke | broj vrijednosti | točnost (%) |
| mfcc | 63 | 48 |
| loudness | 3 | 18,5 |
| mfcc, loudness | 66 | 41,5 |
| nonzero count | 3 | 14,5 |
| spectral centroid | 3 | 17 |
| mean | 3 | 16 |
| mean, nonzero count | 6 | 13 |
| mean, nonzero count, loudness | 9 | 18,5 |
| mean, nonzero count, loudness, mfcc, spectral\_centroid | 75 | 42,5 |
| mean, nonzero count, loudness, mfcc, spectral centroid, spectral slope, spectral standard deviation | 81 | 39 |

Vidi se da više značajki može dati bolje rezultate. Iako niti jedna značajka sama za sebe nije dobra kao mfcc, mfcc uz 4 druge značajke daje bolje rezultate. Moguća je, međutim, i suprotna tendencija što se vidi iz slučaja sa 7 značajki koje daju manju točnost od slučaja s 5 značajki.

### Usporedba s libsvm-om

Uz klasifikator koji smo sami napravili, koristili smo i gotov klasifikator „libsvm“ za usporedbu.

Za oba klasifikatora su korišteni najbolji pronađeni parametri i baza GTZAN . Za naš kreshvm je korišteno C=0.6, polinomni kernel 1. stupnja. Za kreshvm je korišten rbf kernel uz gama=0.1 i C=100 (naredba svm-train -s 0 -c 100 -g 0.1 -v 5). Točnost je određena peterostrukom unakrsnom usporedbom (engl. cross-validation), što smo smatrali najbliže našem dijeljenju skupa na 80% za učenje, 20% za testiranje.

Rezultati usporedbe kreshvm-a i libsvm-a:

|  |  |  |
| --- | --- | --- |
| klasifikator | značajke | točnost |
| kreshvm | sve | 39 |
| libsvm | sve | 60,7 |
| kreshvm | mfcc | 48 |
| libsvm | mfcc | 58,2 |

Vidi se da je libsvm nešto bolji od našeg kreshvm-a, ali ne mnogo. Treba, također, uzeti u obzir da su na libsvmu radili stručnjaci u tom području više godina.

### Veličina skupa za treniranje

Usporedili smo rezultate u ovisnosti o tome koliko je velik bio skup za učenje, tj. koliki je dio korišten za učenje, a koliki za testiranje klasifikatora.

Korištena je GTZAN baza, 80% treniranje, a parametri svm-a su: C=0.6, polinomni kernel 1. stupnja.

Rezultati u ovisnosti o veličini skupa za učenje:

|  |  |
| --- | --- |
| dio cijele baze koji se koristi za učenje (ostatak za testiranje) | točnost (%) |
| 50 | 28,2 |
| 60 | 28,5 |
| 70 | 33,6667 |
| 80 | 39 |
| 90 | 55 |

Vidi se da povećanjem skupa za učenje gotovo jednoliko raste točnost klasifikacije.