ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Systém pro doporučení tanečního stylu
Student:	Tomáš Dejmek
Vedoucí:	Ing. Petr Pulc
Studijní program:	Informatika
Studijní obor:	Teoretická informatika
Katedra:	Katedra teoretické informatiky
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Hudba má většinou poměrně přesně definovatelný žánr. Mapování na taneční styl ale závisí na mnoha dalších faktorech, včetně samotné interpretace konkrétní skladby.

Cílem této bakalářské práce je navrhnout klasifikátor, který nabídne uživateli seznam doporučených tanečních stylů pro konkrétní zvukový záznam.
1. Seznamte se s pokročilými metodami analýzy zvuku a hudby, prozkoumejte deskriptory MPEG-7.
2. Vyberte vhodná data k extrakci, například rytmické vzory.
3. Nalezněte a statisticky podložte rozdíly v datech pro jednotlivé tance.
4. Z klasifikátorů definovaných v SW MATLAB vyberte alespoň dva vhodné.
5. Klasifikátory natrénujte, otestujte na oponentem dodané hudební knihovně a okomentujte výsledky.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 23. ledna 2016

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra teoretické informatiky

Bakalářská práce

Systém pro doporučení tanečního stylu

Tomáš Dejmek

Vedoucí práce: Ing. Petr Pulc

17. května 2016

Poděkování

Chtěl bych poděkovat panu Ing. Petrovi Pulcovi za vedení a konzultace, které mi během práce poskytoval. Dále bych chtěl poděkovat Ing. Janu Trávníčkovi za vymyšlení tématu a poskytnutí své osobní sbírky taneční muziky pro testování výsledků.

V poslední řadě bych chtěl poděkovat mamce, tátovi, babičce, sestře a svým přátelům za průběžnou podporu během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 17. května 2016 .....

Odkaz na tuto práci

Dejmek, Tomáš. *Systém pro doporučení tanečního stylu.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

V této práci se zabýváme rozpoznáváním tanečních stylů z hudebních nahrávek. Představíme si základní rysy standardních i latinskoamerických tanců, které následně budeme hledat v signálu nahrávky. Poměrně značnou část jsme věnovali analýze zvuku a hudby, kde nechybí popis FFT, extrakce základních a spektrálních deskriptorů ani algoritmus pro odhad tempa. Na kapitolu s analýzou navazuje kapitola o klasifikaci, ve které hledáme nejvhodnější klasifikátor a jeho konfiguraci.

Klíčová slova Rozpoznávání, strojové učení, analýza zvuku, FFT, taneční styl, zvuková nahrávka

Abstract

This bachelor thesis deals with the classification of dance styles from music records. The thesis introduces characteristics of ballroom and latin dances, which will be searched in signals of music tracks. A relatively large part of thesis is about audio and music analysis, which contains the definition of FFT, extraction of basics and spectral descriptors and the tempo estimation algorithm. The analysis is followed by a chapter about classification, which focuses on choosing suitable searching classifier and its configuration.

Keywords Classification, mashine learning, musical analysis, FFT, dance style, music track

Obsah

Seznam obrázků

Seznam tabulek

Úvod

V dnešní době strojové učení v oblasti zvuku má za sebou řadu úspěchů a stálé získává nové. Strojové učení v oblasti zvuku jde ruku v ruce s analýzou zvuku, která se zabývá získáváním získáváním informací ze syrových dat zvuku. Extrahované informace používáme například pro filtrování a vyhledávání dat. Můžeme podle nic zjistit zda je obsahem mluvená řeč, hudba nebo například potlesk. Dokážeme zjistit i daleko podrobnější informace jako například kolik mluví mluvčích na nahrávce a určit jsou pohlaví. Skvělých výsledků dosahuje přepis anglicky mluvené řeči do textové podoby, nejznámější z nástrojů, co takový přepis umí, je již několik let nasazený na portálu YouTube, kde si ho můžete sami vyzkoušet. Zajímavé projekty přišly i v oblasti hudby, jako jsou například rozpoznání hudebních nástrojů, převod hudební nahrávky na akordy či noty a rozpoznávání žánrů. Strojové učení ovšem neřeší problém exaktně, musíme počítat s nějakou chybovostí, kterou si můžeme změřit na testovacích datech.

Struktura práce

1. **Cíl práce:** V první kapitole jsme definovali cíle a problém této práce. Dále zde najdeme krátký přehled o podobných pracích, které nám slouží jako zdroj informací.

2. **Popis tanců:** Zde najdeme základní charakteristiky standardních i latinskoamerických tanců.

3. **Analýza zvuku:** V této kapitole si ukážeme, jak je zvuk reprezentovaný v počítačích. Definujeme a vysvětlíme si diskrétní Fourierovu transformaci, kterou budeme používat pro vytvoření frekvenčního spektra a spektrogramu, pomocí kterých si ukážeme některé základní prvky zvuku.

4. **Standard MPEG-7:** V této kapitole se budeme zabývat extrakcí parametrů pro klasifikaci zvuku, nechybí ani algoritmus pro odhad tempa.
5. **Vytvoření datasetu:** Zde jsme popsali způsob, jakým jsme vytvořili dataset, na kterém jsme provedli základní statistiku dat. Speciální pozornost jsme věnovali parametru tempo, u kterého jsme si udělali analýzu kvality jeho odhadu a popsali jeho informační přínos pro klasifikaci.
6. **Klasifikace:** V této kapitole jsme vytvořili několik klasifikačních modelů, které jsme potom optimalizovali pro klasifikační přesnost.
7. **Závěr:** V závěru jsme shrnuli celou práci a popsali představy, jak by se dalo na práci navázat.

Kapitola 1

Cíl práce

Cílem práce je navrhnout a implementovat klasifikátor, který k dané hudební nahrávce dodá seznam doporučených tanečních stylů. Vyzkoušíme několik klasifikátorů a vybereme ten nejúspěšnější. Pro měření úspěšnosti použijeme křížovou validaci.

1.1 O problému

Tento problém spojuje dva dílčí problémy dohromady. Jedním z nich je analýza obsahu zvuku, kde bude hlavním cíle získat vhodné atributy z rozmanitých dat. V datech máme různé hudební nahrávky, které jsou v odlišných formátech zakódované s různými frekvencemi a ještě s odlišným počtem zvukových kanálů. V oblasti analýzy zvuku a hudby probíhají aktivně výzkumy od začátku sedmdesátých let. Prozkoumáme možnosti, které nám analýza zvuku nabízí a relevantní z nich vybere. Tím se budeme zabývat v druhé kapitole: Analýza zvuku.

Potom co si připravíme vhodný dataset pomocí analýzy zvuku, úloha se nám zredukuje na klasický klasifikační problém. Představíme si vhodné klasifikátory a vytvoříme klasifikační model, ve kterém budeme testovat několik vybraných klasifikátorů, budeme hledat jejich ideální konfigurace a porovnáme si výsledky.

1.2 Motivace

Pokud by se povedlo dosáhnout dobrých výsledků v klasifikaci tanečních stylů, byl by to první krůček k hned několika zajímavým projektům, kterými by se dalo navázat. Například mobilní aplikace pro rozpoznávání, mobilní telefon by chvilku mikrofonem odposlechl vzorek muziky a vyhodnotil by výsledek. To by ocenili tanečníci a tanečnice, jež mají s rozpoznáváním stylů ještě potíž. Archivy s taneční muzikou by mohl být filtrován automaticky bez otravného

ručního třídění, to by mohli ocenit správci hudebních archivů, organizátoři tanečních soutěží nebo členi tanečních klubů. Roboti by se mohli naučit tančit a jejich autoři by mohli soutěžit o nejlepší robotický taneční pár.

1.3 Současný stav řešení

Nepodařilo se nám nalézt žádnou práci, která by se zabývala rozpoznáváním tanečních stylů z hudebních nahrávek v oblasti strojového učení. Můžeme tedy předpokládat, že ekvivalentní práce neexistuje. Našli jsme práce o rozpoznávání žánrů, což je principiálně velice podobné. Vědecká skupina Mir z Vídeňské technické univerzity již z multimédii běžně pracuje [1]. Pro analýzu zvuku používajízákladní dekriptory z MPEG-7 a některé pokročilé například rytmické vzorky. Pro klasifikaci použili samoorganizující se mapy. Další zdroje již nejsou příliš podrobné [2, 3, 4].

Kapitola 2

Popis tanců

Každý taneční styl je definován svými pravidly. V této kapitole se podíváme na jejich základní charakteristické rysy. Tance se člení na standardní, latinskoamerické a ostatní. Seznam tanců, které jsme zařadili do této kapitoly je omezen na tance, které bude systém rozpoznávat.

2.1 Standardní tance

Standardní tance jsou charakteristické uzavřeným párovým držením. Jejich původ je z evropských tradic až na tango, které pochází z Uruguaye a před svojí standardizací bylo dlouho považované za latinskoamerický tanec. Pro tyto tance kromě tanga je charakteristický švihový pohyb. Standardní tance jsou historicky starší oproti latinskoamerickým tancům a vyžadují přísnější požadavky pro volbu oděvu. Tanečník by měl mít frak nebo vestu a jeho partnerka dlouhé šaty.

Valčík

Valčík je postupový tanec ve tříčvrťovém taktu, vyvinul se v rakouských Alpách z rakouského landleru. Tempo valčíku je přibližně 175 úderů za minutu. Neformálně se valčík označuje jako „král všech tanců“. Zajímavostí je, že tento tanec byl první tanec, který se tančil v těsném držení a církev ho zakázala jako tanec zdraví škodlivý a hříšný [10].

Waltz

Waltz je tanec v tříčvrťovém taktu, vyvinul se v Anglii zkombinováním prvků amerického tance boston a landleru, který je předchůdcem valčíku. Jeho rytmus klade důraz na první dobu a má tempo přibližně 80 úderů za minutu.

Tango

Tango je tanec v 2/4 nebo 4/8 rytmu s volným důrazem, který má svůj původ v Uruguayi. Tempo tanga je kolem 128 úderů za minutu. Dovolím si přiložit citaci, která nepřesně popisuje další hypotetický rys, který charakterizuje tango. „Ostré otáčení partnerčiny hlavy jen podkresluje napětí a energii, kterou tango nabízí.“ [12] Neformálně by se dalo říct, že pro tango jsou typické ostré přechody v rytmu.

Slowfox

Slowfox nemá své základy v lidových tancích a na rozdíl od ostatních standardních tanců vznikl uměle. Jeho jeho rytmus je posazen ve 4/4 taktu s tempem přibližně 126 úderů za minutu a klade důraz na první a třetí dobu.

Quickstep

Tanec quickstep se vyvinul ze slowfoxu, a proto se mu také říká rychlý foxtrot. Jeho tempo 200 až 208 úderů za minutu z něho dělá nejrychlejší standardní tanec.

Tabulka 2.1: Charakteristiky standardních tanců [3].

-	Waltz	Tango	Valčík	Slowfox	Quickstep
Takt	$\frac{3}{4}$	$\frac{4}{8}$ nebo $\frac{2}{4}$	$\frac{3}{4}$	$\frac{4}{4}$	4/4
Tempo [takt./min.]	28-30	31-33	58-60	28-30	50-52
Metronom [BPM]	84-90	124-132	174-180	112-120	200-208
Důraz v taktu	1	volný	1	1,3	-

V tabulce si povšimneme, že tance valčík a waltz jsi jsou velice podobné, mají stejný tak a oba mají důraznou první dobu, ale dokážeme je rozlišit tím, že valčík má přibližně dvojnásobnou rychlost.

2.2 Latinskoamerické tance

Pro latinské tance je typické volné držení páru, kdy partner předvádí partnerku, která má mnohdy složitější kroky a více se otáčí. U ženy je může být volba šatů odvážnější oproti standardním tancům a jejímu je povolena samotná košile.

Samba

Je tanec pocházející z Brazílie, který se objevil v Evropě ve dvacátých letech dvacátého století. Samba má rytmus v 2/4 anebo 4/4 taktu a tempo přibližně 104 až 108 úderů za minutu.

Rumba

Rumba je velice intimní kubánský tanec charakteristický pomalou hudbou [11]. Její rytmus je v 4/4 taktu s důrazem na čtvrtou dobu. Tempo rumby je přibližně 100-108 úderů za minutu.

Cha-cha

Cha cha je původem z Kuby a její základ vychází z rumby. Hudba je ve 4/4 taktu s důrazem na první dobu. Hudba má 4/4 rytmus a tempo 120 až 128 úderů za minutu. Její melodie vyvolává pocity vesela a bezstarostnosti. Mezi další znaky patří půlená čtvrtá doba, která je typická výhradně pro cha chu.

Jive

Jive je swingový dynamický tanec ovlivněný rokenrolem. Jeho rytmus je v 4/4 taktu a klade důraz na druhou a čtvrtou dobu. Jive má tempo přibližně 172 úderů za minutu.

Salsa

Salsa má 4/4 rytmus, a existuje ve dvou verzích rytmu (cowbell rhythm nebo conga rhythm), důrazné doby můžou být 1., 3., 5. a 7. nebo 2., 4., 6. a 8.

Tabulka 2.2: Charakteristiky latinskoamerických tanců [3].

-	Cha cha	Jive	Rumba	Samba	Salsa
Takt	$\frac{4}{4}$	$\frac{4}{4}$	$\frac{4}{4}$	$\frac{2}{4}$	$\frac{4}{4}$
Tempo [takt./min.]	30-32	42-44	25-27	50-52	37-60
Metronom [BPM]	120-128	168-176	100-108	100-104	148-240
Důraz v taktu	1	2, 4	4	2	různý

2.3 Ostatní tance

Polka

Společenský tanec v 2/4 taktu, který vznikl okolo roku 1830 v Čechách. Tempo polky má velké rozpětí 88 až 126 úderů za minutu.

Mazurka

Tento lidový tanec je původem z Polska. Jeho rytmus je posazen do 3/4 taktu s tempem přibližně 141 úderů za minutu.

Tabulka 2.3: Charakteristiky ostatních tanců [3].

-	Polka	Mazurka
Takt	$\frac{2}{4}$	$\frac{3}{4}$
Tempo [takt./min.]	44-63	46-48
Metronom [BPM]	88-126	138-144
Důraz v taktu	obě	1

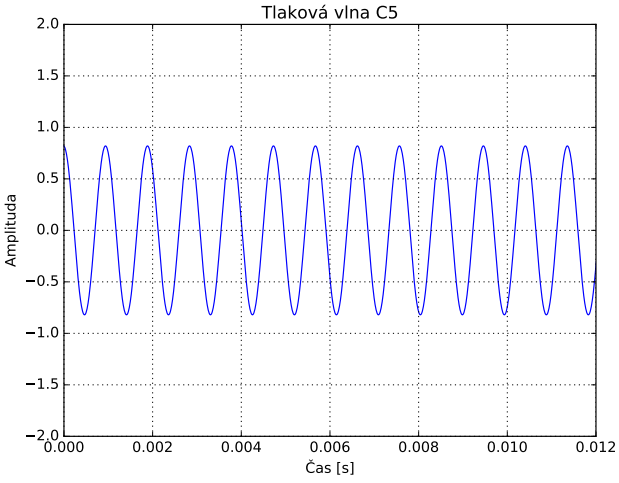
2.4 Poznámka

Tance slowfox, quickstep a mazurka jsme v této práci nezpracovali pro klasifikaci.

Kapitola 3

Analýza zvuku

Zvuk z pohledu fyziky je vibrace o frekvenci v rozsahu 19 až 20000 Hz (lidsky slyšitelné frekvenci), která vychází ze svého zdroje a je přenášená v pevném, kapalném nebo plynném materiálu. Vibrace má podoby mechanického vlnění, podélného vlnění a také podobu tlakové vlny. Zvuk v podobě tlakové vlny střídavě zvyšuje a snižuje tlak materiálu ve fixní vzdálenosti od zdroje. Zvuk můžeme reprezentovat jako funkci jedné proměnné, která má na ose x čas a na ose y je hodnota vychýlení, která má vliv na relativní tlak vůči tlaku materiálu, ve kterém vibrace proudí. Reprezentujme tedy zvuk jako funkci $f(x) : \mathbb{R} \to \mathbb{R}$ pomocí které budeme zvuk analyzovat, především prvku hudby. Taková funkce nemusí být spojitá. Spojitost nám může narušit například výbuch, který vytvoří dočasně vakuum okolo zdroje. V této části si jenom opravdu krátce a



Obrázek 3.1: Elektronicky generovaná nota C5

stručně představíme *diskrétní Fourierovu transformaci* (DFT) a předtím si

řekneme něco *Fourierových řadách.*

3.1 Fourierova řada

Josephov Fourier (1768-1830) prohlásil v roce 1822, že některé funkce je možné vyjádřit jako součet nekonečné řady harmonických funkcí a ukázal to na několika příkladech. V té době ještě matematika neměla zcela pevně stanovené některé pojmy a nebylo přesně zjištěno pro jaké funkce tvrzení platí. Velcí matematici v devatenáctém století vybudovali celou teorii trigonometrických řad, při snaze objasnit hypotézu a dodnes jsou v problematice nevyřešené otázky.

Definice. *Každá reálná funkce F jedné reálné proměnné, která je periodická s periodou $2T$ a integrovatelná na intervalu $(-T, T)$ je vyjádřitelná jako nekonečná trigonometrická řada.*

$$F(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos(\tfrac{\pi k x}{P}) + b_k \sin(\tfrac{\pi k x}{P}) \right), \text{ pro } \ x \in \mathbb{R}.$$

Parametry $a_0, a_1, ..$ a $b_1, b_2, ..$ lze získat potom vztahem následujícím vztahem.

$$a_k = \frac{1}{T} \int_{-T}^{T} F(x) \cos(\frac{\pi k x}{T}) dx, \text{ pro } \ k \in \mathbb{N}_0.$$

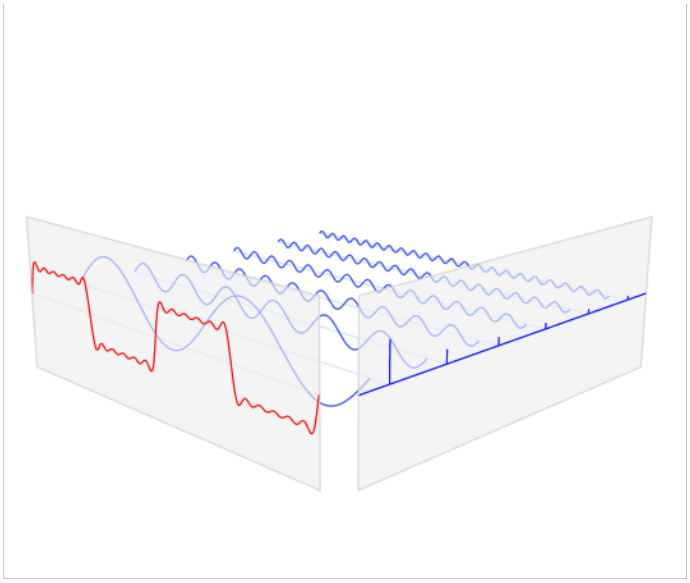
$$b_k = \frac{1}{T} \int_{-T}^{T} F(x) \sin(\frac{\pi k x}{T}) dx, \text{ pro } \ k \in \mathbb{N}.$$

Funkce F může mít na intervalu $(-T, T)$ konečný počet bodů nespojitosti, ale nesmí mít bod nespojitosti bez existence limity v něm, jinými slovy může obsahovat takové body nespojitosti, které lze dodefinovat limitou.

Poznámky

- Funkce, které nejsou periodické můžeme vyjádřit stejným způsobem ovšem pouze na intervalu $(-T, T)$, mimo interval se bude vyjádření chovat periodicky.
- Vyměníme-li nekonečno za konečnou proměnnou N dostáváme aproximaci funkce F s rostoucím N klesá chyba.
- Fourierovy řady mají slabší podmínku, než *Taylorův polynom*, není třeba všech derivací v bodě do řádu n, dokonce ani není nutní spojitost. Někdy je sestavení Fourierovy řady výpočetně snadnější.

Pro představu co vlastně výpočet vektorů a a b dělá a co to má vlastně společného s analýzou hudby se podíváme na následující obrázek 3.2. Vlevo je červeně zobrazena funkce F, to bude náš zvuk s časem na ose x a hodnotou



Obrázek 3.2: Znázornění furierové řady

signálu na ose y. Pro tuto naší funkci hledáme vektory a a b, postupně integrujeme pro každou další složku. Za funkcí jsou vygenerované sinusové funkce, které jsou násobené takovými hodnoty aby, když při celkovém sečtení všech jsme dostali zpět tvar funkce F, prvek a_0 je absolutní člen, který celou funkci posouvá po ose y. Nejzajímavější pro nás jsou hodnoty $a_1, a_2, ..$ a $b_1, b_2, ..$, které ve kterých je přesně promítnuto jaké frekvence jsou ve signálu obsaženy. V případě na obrázku vstupní signál byla zřejmě lichá funkce, kosinus je sudá funkce a sudá funkce krát lichá funkce je lichá funkce.

Raimanův integrál z liché funkce na intervalu $(-T, T)$ vyjde vždy nula. Tedy a je nulový vektor a proto nemáme v obrázku žádnou kosinovou funkci. Vpravo je graf, který na ose x má frekvence a na ose y intenzitu s jakou je frekvence obsažena. Přesněji řečeno pro x rovno nule máme číslo $y = a_0$ a pro každé kladné x máme na y dvojici reálných čísel a_i a b_i, a_i popisuje výraznost kosinové vlny a b_i značí výraznost sinové vlny při té samé frekvenci.

V analýze zvuku nám nezáleží na tom jestli hraje sinusová vlna nebo kosinová, je to jen posunutí o čtvrt periody. Co je pro nás důležité je zjištění množství energie na dané frekvenci, brzy si povíme proč. Pro výpočet celkové intenzity můžeme použít triviální převod pomocí Pythagorové věty $\sqrt{(a_i^2 + b_i^2)}$. Dvojice a_i a b_i můžeme vnímat jako komplexní číslo z_i a potom intenzitu získáme jako $|z_i|$.

3.2 Diskrétní Fourierova transformace

Fourierova řada je výborná věc, ovšem v analýze digitálního zvuku je několik detailů, které nás od jejího použití odrazují. První problém je převedení digitálních dat do spojité funkce, máme povoleno pouze konečné množství bodů nespojitosti. Digitální data nám poskytnou konečné množství hodnot, v praxi je běžné obsažení 44 100 definovaných bodů v jedné sekundě záznamu. Pokud bychom takovou funkci získali, další problém by byl ji integrovat. Proto vznikla diskrétní Fourierova transformace (DFT), která jako vstup přijímá N pravidelně vzdálených bodů z signálu a provede nám to stejnou transformaci na vstupní množině.

Definice (Diskrérní Fourierova transformace)**.** *Mějme vektor komplexních čísel $a = (a_0, a_1, .., a_{n-1}) \in \mathbb{C}^n$ popisující hodnoty signálů po pravidelných časových úsecích, potom jeho diskrétní Fourierovou transformací je vektor $A \in \mathbb{C}^n$.*

$$A_k = \sum_{m=0}^{n-1} a_m exp(-2\pi i \frac{mk}{n}), \text{ pro } \; k=0,1,...,n-1$$

A zpětnou transformaci potom definujeme následovně.

$$a_m = \frac{1}{n} \sum_{k=0}^{n-1} A_k exp(2\pi i \frac{mk}{n}), \text{ pro } \; k=0,1,...,n-1$$

Zbývá ještě vyřešit poslední nešvár, abychom ji mohli začít používat a tím je složitost. Naivní implementace DFT (3.1), tedy přepsáná definice do programovacího jazyka, má složitost $\theta(N^2)$ při hustotě dat 44 100 hodnot na sekundu záznamu, začínáme mít problém s časem už při převedení pětisekundového vzorku útržku záznamu.

Listing 3.1: DFT naivně $\theta(n^2)$

```
def DFT(a):
    n = len(a)
    A = [0] * n
    for k in range(n) :
        for m in range(n):
            A[k] += a[m] * exp(-2 * pi * 1j * m * k / n)
    return A
```

3.3 Rychlá Fourierova transformace

Naštěstí pánové James Cooley a John Tukey z IBM při společném výzkumu v roce 1965 vyvinuli rekurzivní implementaci algoritmu DFT, která má složitost $\theta(n \log n)$ známá jako Cooley-*Tukey Fast Fourier Transform* (FFT). Tento algoritmus potřebuje velikost vstupu mít ve tvaru mocniny dvojky $n = 2^N$, vstupy se typicky doplňují nulami na potřebnou velikost.

Listing 3.2: Cooley–Tukey FFT $\theta(n \log n)$

```
def FFT(x):
    N = x.shape[0]
    if N <= 2:
        return [x[0]+x[1], x[0]+exp(-2j*pi*1j)*x[1]]

    X_even = FFT(x[::2])
    X_odd = FFT(x[1::2])
    factor = exp(-2j * pi * arange(N) / N)
    return concatenate([X_even + factor[:N / 2] * X_odd,
                        X_even + factor[N / 2:] * X_odd])
```

3.4 Frekvenční spektrum

Frekvenční spektrum signálu F je zobrazení velikostí amplitud *harmonických frekvencí*, které když se sečtou dají dohromady signál F, do *frekvenčního pásma*. Jinými slovy ke každé frekvenci v nějakém rozsahu určujeme, jak se podílí na signálu.

Nyní si ukážeme kde ve výstupu FFT najdeme frekvenční spektrum. Získaný n-prvkový vektor komplexních čísel $a \in \mathbb{C}^n$ získaný FFT vektoru $A \in \mathbb{C}^n$ pro $n = 2^k, k \in \mathbb{N}$ můžeme rozdělit na tři části, které jsou první prvek, sekvence prvků začínající druhým až $n/2+1$ a zbytek. První prvek má na reálné složce nultého prvku absolutní člen d, který je pouze posun $Re(a_0) = d$ a na imaginární má nulu $Im(a_0) = 0$. Druhá část nám tvoří hodnoty frekvenčního spektra ($|x|, x \in \{a_1, a_2, ...a_{n/2}\}$). Třetí část zrcadlí druhou část okolo prvku $a_{n/2}$ akorát při zrcadlení došlo k prohozením znaménka u imaginárních složek, tedy platí $a_{n/2-i} = \overline{a_{n/2+i}}$, pro $i = 1, ..., \frac{n}{2} - 1$. Zrcadlení je způsobeno tím, že DFT počítá i záporné frekvenční pásmo, které je symetrické s kladným.

3.5 Spektrogram

Spektrogram je grafické znázornění frekvenčních spekter během času signálu. Definuje se velikost okénka tvaru mocniny dvojky a překryv který menší než velikost okénka. Zvukový signál se navzorkuje do okének s překryvem a následně se pro každé okénko provede FFT a výsledek se zanese do grafu, ve kterém obvykle teplé barvy označují výrazné frekvence. Na ose x je čas a na ose y je frekvenční pásmo.

Harmonické frekvence

Mějme fundamentální frekvenci f, její i-tá harmonická frekvence $(i)f$ pro $i = 1, 2, 3...$ Tedy například k frekvenci 440 Hz je první harmonická frekvence 440 Hz, druhá 880 Hz, třetí 1320 Hz a podobně. Frekvence, které nejsou

harmonické nazýváme enharmonické. Některé zdroje uvádí první harmonickou frekvenci jako druhý násobek a ostatní, že první harmonická frekvence je rovna fundamentální frekvenci, dohodneme se na druhé variantě.

3.6 Hluky, tóny a noty

Zvuky se dělí na *tóny* a *hluky*, přičemž tón na rozdíl od hluku má pevně danou svojí výšku. Noty popisují tóny, definují jejich výšku a trvání. Tón navíc od noty má ještě svojí hlasitost a barvu.

- *Výška tónu* (pitch) Lze definovat jako nejnižší frekvence, která se v tónu nachází. Z nahrávky, na které je pouze jeden zahraný tón, můžeme výšku tónu spočítat tak, že na křivce zvuku si najdeme extrémy v pozitivní nebo negativní polovině, které se pravidelně opakují. Jejich časová vzdálenost je perioda T sekund, tedy frekvence $f = \frac{1}{T}Hz$. Pokud máme více tónů zahraných dohromady, tento přístup nám bude fungovat na hledání nejnižší frekvence. Lépe je výška tónu vidět ve frekvenčním spektru nebo spektrogramu.

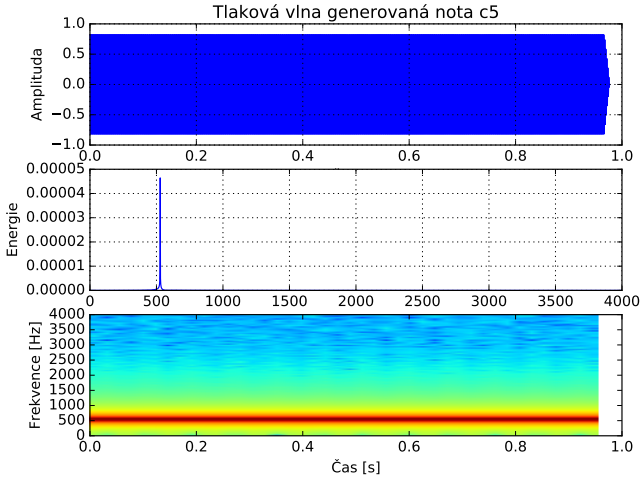
- *Trvání tónu* (duration) Trvání doba jakou tón hraje lze vidět ve spektrogramu.

- Hlasitost (loadness příp. také *intenzita* nebo *energie*) se dá získat ze spektrogramu sečtením intenzit všech harmonických frekvencí.

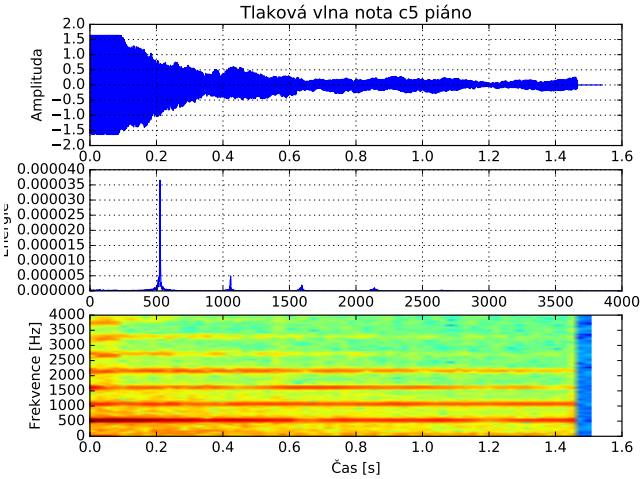
- *Barva tónu* (timbre) Lze popsat jako harmonické a enharmonické frekvence, které s nejspodnější frekvencí hrají. Každý typ hudebního nástroj hraje tóny o jiných barvách. Dokonce i stejné typy hudebních nástrojů můžou hrát jiné barvy, například takový klavír může hrát jasně, tmavě, teple a tvrdě, na jeho barvu má vliv z jakého je dřeva je vyrobené, jeho tvar a jaké struny používá. U dechových nástrojů barvu velice ovlivňuje muzikant svým dechem. U strunných pak pozice hrající ruky nebo smyčce vůči ukotvení strun

  Ve spektrogramu jsou harmonické frekvence, které se s tónem pojí, dobře vidět.

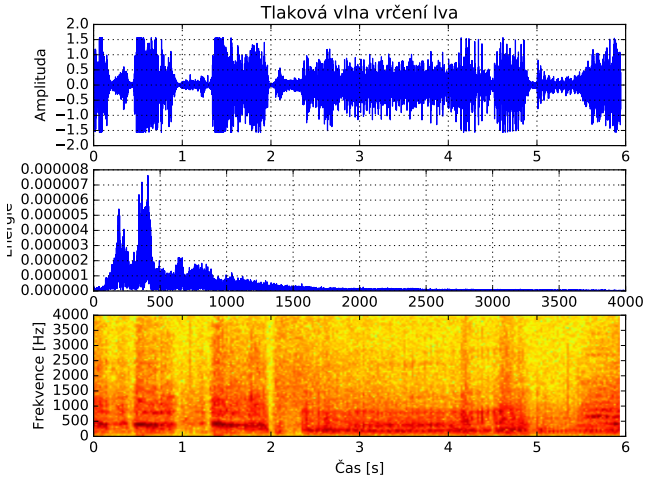
Popsali jsme si 3 způsoby jakými můžeme popsat zvuk a vizualizovat ho a vysvětlili jsme si, co je nota, tón a hluk. Na následujících třech obrázcích si ukážeme, jak vidíme rozdíly v jejich zobrazení. Prvním obrázkem je uměle vygenerovaná nota C5, druhým je ta samá nota zahraná na klavír a třetí je řev tygra tedy hluk. Nota C5 má frekvenci 523 Hz, pokud nastavíme referenční notu A4 na frekvenci 440 Hz, což obvykle bývá. Generovaná nota má pouze jednu naprosto čistou frekvenci, takové čisté tony se v běžném světě obvykle nevyskytují.



Obrázek 3.3: Generovaná nota C5



Obrázek 3.4: Nota C5 hraná na klavír, na rozdíl od vygenerovaného tónu zde vidíme 4 harmonické frekvence a dokonce v prvních 200 ms jich je 7.



Obrázek 3.5: Řev lva je typický hluk, ve spektrogramu není vidět žádná čistá frekvence

Kapitola 4

Standard MPEG-7

Předtím než se pustíme do matematických popisů *deskriptorů* z *MPEG-7* si v této kapitole krátce představíme *standardy MPEG* a něco málo si řekneme o tom co přinesly. Standardy MPEG vytváří stejnojmenná skupina specialistů *Moving Picture Experts Group*, jenž byla založena v roce 1988, kdy začali čelit výzvě zakódování digitálního videa s přidruženým zvukem o průtoku maximálně 1,5 Mbps. Výsledkem byl od roku 1993 celosvětově uznávaný ISO-IEC standard *MPEG-1* o komprimaci, ukládání a přenosu audio a video dat, který přinesl mimo jiné i dodnes používaný formát *Layer 3* známý jako MP3. V roce 1995 se stal ISO-IEC standardem *MPEG-2*, který přináší lepší kompresi a umožňuje zakódovat HD video signál do průtoku 15 Mbps. V roce 1999 přichází MPEG-4 s lepší kompresí než MPEG-2, dokáže HD video signál zakódovat do průtoku 6 Mbps a navíc přináší standard pro reprezentaci 3D modelů. Dále tato skupina vyvinula ještě několik standardů MPEG-7, *MPEG-21* a několik dalších, nejnovějším je *MPEG-H*, který si bere za úkol například zlepšit kvalitu 3D ozvučení domácích kin a zvýšit kvalitu komprese.

Standardy MPEG-7 se od ostatních standardů z rodiny MPEG liší především tím, že neslouží ke kompresi ani zakódování AV obsahu. MPEG-7 představuje rozhraní pro popis obsahu široké škály multimediálních dat, jako jsou obrázky, video, zvuk, 3D modely, animace a další. Standardizuje několik deskriptorů pro popis AV dat a obecné principy popisování dat, které slouží pro rychlé vyhledávání a filtrovaní, případně shlukování podobných dat. Deskriptory jsou navrženy, tak aby byly nezávislé na reprezentaci dat a mohli se z dat vypočítat bez potřeby uživatelské interakce.

MPEG-7 představil DDL, což je *popisovací jazyk*, který je založený na bázi XML. Také přestavil DSs, to jsou *popisovací schémata*, která popisují vztahy mezi atributy. Ve schématech jsou můžou být jak automaticky vypočítatelné hodnoty, tak lidsky zapsaná *metadata*. Myšlenka tohoto systému spočívá v tom, že požaduje-li nějaký nástroj nějakou hodnotu nejprve se podívá nástroj do metadat, zda tam takové informace nejsou. Pokud ne, požadované informace si vypočítá z dat, pokud to jde. A informaci může uložit mezi me-

tadata, a to nejlépe na místo určené schématem. Metadata zapsaná v DDL se buďto zapisují v komprimované podobě přímo do multimediálních souborů nebo jsou ukládány mimo.

4.1 Deskriptory MPEG-7 Audio

Algoritmy pro feature extraction slouží k analyzování obsahu nahrávky, především hudby. Zatímco některé základní atributy jsou získávány přímo ze signálu, většina atributů se počítá ze frekvenčního spektra. Vstupní nahrávku si rozdělíme na rámce o dané velikosti a analyzujeme každý rámec zvlášť. Zavedeme si označení M pro velikost rámce a F_i pro signál i-tého rámce.

Zero crossing rate (ZCR)

Zero crossing rate je triviální atribut získaný přímo ze signálu, jeho hodnota specifikuje u signálu počet překřížení nuly během rámce, tedy kolikrát signál přešel signál z pozitivních hodnot do negativních nebo z negativních hodnot do pozitivních a je normalizovaný na velikost rámce. Matematicky lze definovat následující formulí.

$$ZCR(i) = \frac{1}{2M} \sum_{k=1}^{M} |sgn(F_i(k)) - sgn(F_i(k))|$$

Kde použitá funkce signum je definována takto.

$$sgn(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

V tomto atributu se odráží míra šumu v nahrávce, je známo, že se stoupajícím množství šumu ZCR roste. Rámce, které obsahují řeč mívají ZCR vyšší oproti rámcům s hudbou [6]. Některé změny, které jsou vidět ve spektrogramu se do ZCR odráží, například přidáním nejnižší frekvence s vysokou amplitudou do zvlněného signálu ZCR dočasně snižuje. Pro jeho jednoduchý výpočet a informačního hodnotu se často používá pro detekci řeči a klasifikaci žánrů.

Energie

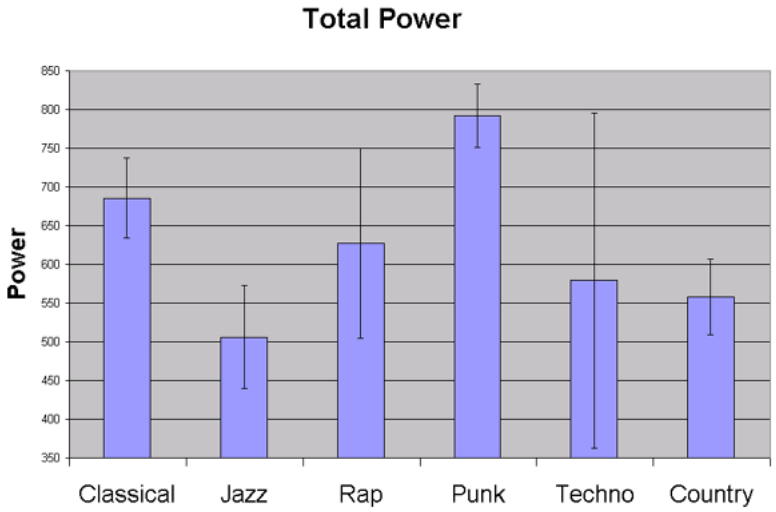
Energie je ukazatel kolik jakou intenzitu zvuk nabývá v daném čase, v literatuře ho často objevuje také pod názvem *power of signal*. Definován je jako součet čtverců z hodnot signálu, dělený velikostí rámce.

$$E(i) = \frac{1}{M} \sum_{k=1}^{M} |F_i(k)|^2$$

Zajímavostí je, že tento atribut také můžeme vypočítat z frekvenčního spektra a to takto. Nechť vektor $A_i = FFT(F_i) \in \mathbb{C}^M$ potom

$$E(i) = \frac{1}{M^2} \sum_{k=1}^{M} |A_i(k)|^2$$

. Proč platí rovnost si dokazovat nebudeme, protože by to bylo nad rámec naší práce. Ačkoliv to na první pohled není zřejmé, tento atribut je velice ceněný. Obvykle hudební nástroje vydávající silnější harmonické frekvence se podílí větším množstvím energie. Do atributu se odrazí množství hudebních nástrojů, jejich hlasitosti a intenzity v harmonických frekvencích. Proto je to užiteční atribut pro klasifikaci žánrů [5].



Obrázek 4.1: Klasifikace žánru podle energie [5]. Na obrázku je vidět, že například punk, klasická hudba a country se dají dobře odlišit podle energie. Naopak techno a rap mají příliš velikou *směrodatnou odchylku*.

Pokud je v nahrávce řeč a analyzovali malé rámce například 25 milisekund, často by se nám střídaly rámce s malou a velkou energii, podrobně o tom povykládá zdroj [4].

Entropie energie

Entropie energie vyjadřuje míru proměnlivosti energie v jednom rámci. Algoritmus výpočtu využívá výše definovanou celkovou *energii* rámce $E(i)$, poté si rozdělí rámec na K malých rámců a každému spočte energii $E_subFrame_k$. Poté získáme dílčích energii dělených celkovou energii.

$$e_k = \frac{E_{subFrame_k}}{E(i)}$$

Entropie je dána vztahem

$$H(i) = -\sum_{k=1}^{K} e_k \log_2(e_k)$$

4.2 Spektrální atributy

Následující atributy budou získávány ze frekvenčního spektra $A_i = FFT(F_i) \in \mathbb{C}^N$, kde $N = \frac{M}{2}$. Velice často nás budou zajímat pouze magnitudy $B \in \mathbb{R}^N$, které formálně definujeme vztahem $B_i = |A_i|$.

Spectral centroid

Název byl vychází z toho, že nám připomíná těžiště frekvenčního spektra. Je získaný vztahem

$$C_i = \frac{\sum_{j=1}^{N} jB_i(j)}{\sum_{j=1}^{N} B_i(j)}$$

v podstatě se jedná o *střední hodnotu* neboli *první moment* ze statistiky, protože pravděpodobnost hodnoty k si můžeme vyjádřit jako

$$P(k) = \frac{B_i(k)}{\sum_{j=1}^{N} B_i(j)}$$

potom vidíme známý vzoreček

$$Expected(B_i) = \sum_{j=1}^{N} x_j P(j), \text{ zde } \; x_j{=}j$$

Jako parametr nám říká o výšce a barvě zvuku, čím vyšší hodnota, tím jsou tóny vyšší. Jasnější nebo studenější tóny budou mít vyšší *spectral centroid*.

Spectral spread

Spectral spread je potom odmocněný *druhý centrální moment*, tedy odhad směrodatné odchylky.

$$S_i = \sqrt{\frac{\sum_{j=1}^{N}(j - C_i)^2 B_i(j)}{\sum_{j=1}^{N} B_i(j)}}$$

Popisuje nám jak je spektrum rozdělené kolem svého těžiště.

Spectral entropy

Je *entropie spektrální energie*, počítá se stejně jako entropie energie až na to, že použijeme žijeme jiný vektor a výsledek vydělíme N. Máme velikost podrámečku R, spočteme si pro každý podrámeček energii ze spektra.

$$j \in (1, 2, ..., \lceil N/R \rceil), end_j = min((j+1)R - 1, N),$$

$$sf_j = \frac{1}{N(end_j - iR)} \sum_{k=iR}^{end_j} (B_i(k))^2$$

Již jsme si řekli, že energie signálu a energie spektra signálu vyjdou stejně.

$$p_j = sf_j / E(i)$$

$$SH(i) = -\sum_{j=1}^{\lceil N/R \rceil} p_j \log(p_j)$$

Maximální entropie dosáhne bude-li energie rozložena rovnoměrně přes všechny frekvence jako je bílý šum. Vysokých hodnot nabývá pro řeč, nižších pro hudbu, ještě nižších pro jednoduché tóny a nula pro ticho.

Spectral flux

Spektrál flux vyjadřuje míru změn dvou sousedních rámců. Počítáme ho jako součet čtvercových vzdáleností přes všechny složky vektoru. Magnitudy frekvenšního spektra se nejprve normalizují podle formule.

$$NB_i(k) = \frac{B_i(k)}{\sum_{j=1}^{N} B_i(j)}$$

Potom je spectral flux definován následovně.

$$SF(i, i-1) = \sum_{k=1}^{N} (NB_i(k) - NB_{i-1}(k))^2$$

Spectral rolloff

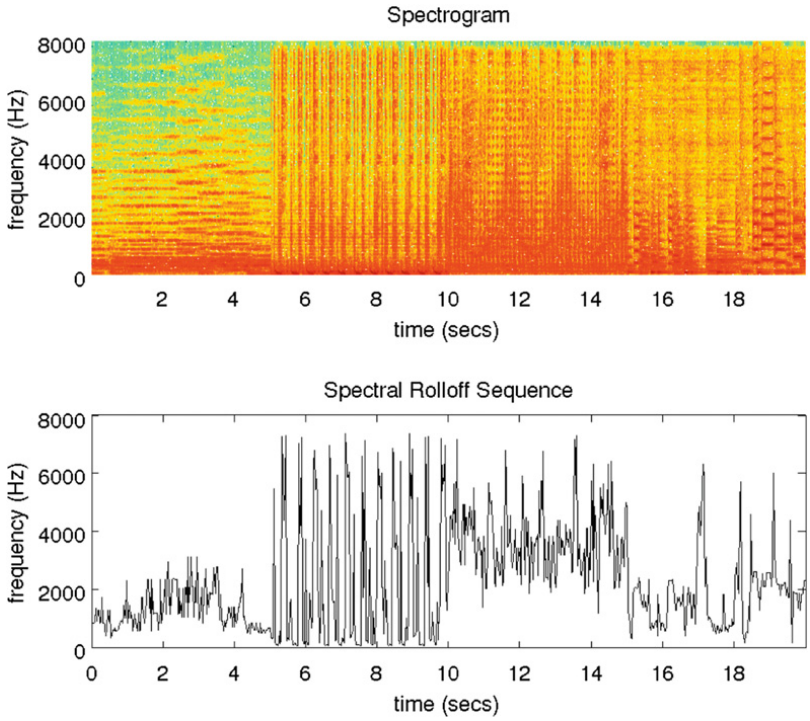
Spectral rolloff je další parametr, který popisuje tvar rozložení frekvenčního spektra. V podstatě vyjadřuje v kolika procentech spodní části spektra se skrývá požadované procento celkového součtu. Před jeho výpočtem si zvolíme $0 < C < 1$ (obvykle se volí hodnota okolo 90 %) a potom hledáme nejnižší m_i, pro které platí.

$$\sum_{j=1}^{m_i} B_i(j) \geq \sum_{j=1}^{N} B_i(j).$$

Spectral rolloff je potom definován jako m_i normalizovaná vůči N.

$$SR(i) = \frac{m_i}{N}$$

Následující obrázek 4.2 zobrzuje jak se mění spectral rolloff v průběhu času na nahrávce skládající se ze čtyř pětisekundových úseků. Prvních na prvním úseku je klasická hudba, další dva úseky jsou z dvou různé elektronických skladeb a poslední úsek je jazzová muzika.



Obrázek 4.2: Spectral rolloff pro na čtyřrch úsecích ruzných nahrávek, postupně klasická muzika, dvakrát elektronická hudba a jazzová muzika.

4.3 Mel-Frequency Cepstrum Coefficients

Mel-Frequency Cepstrum Coefficients (MFCCs) je L-členný vektor, který skvěle popisuje lidskou řeč i muziku. Již byl využit například rozpoznávání řečníka, jazykový obsah, emoce [7], ale i klasifikaci hudby [5]. Zjednodušeně řečeno složky vektoru popisují rozložení energie ve frekvenční pásu, ovšem hodnoty jsou přeškálovány, tak jak je slyšíme. Pokud si na pianu budeme přehrávat noty, tak jak jsou po sobě, změnu mezi tóny vnímáme lineárně, jinými slovy připadá nám, že další nota se stále zvyšuje o stejný kus. Ve skutečnosti, ale skok o oktávu zdvojnásobí výšku tónu a skok o i oktáv udělá 2^i násobek původní výšky. Z toho je zřejmé, že zvuky vnímáme logaritmickým vztahem.

Podobně to tak je i s hlasitostí, se stoupající energií nám hlasitost h roste vztahem přibližně $\theta(\sqrt[3]{h})$. Oba tyto aspekty máme zakomponované ve výpočtu MFCCs, který je sice dlouhý na popis, ale není příliš složitý na implementaci ani pochopení. Výpočet rozdělíme na následující kroky:

1. Spočteme *periodogram* frekvenčního spektra
2. Vytvoříme *Melovy filtry* a sečteme energii v každém filtru
3. Energii si logaritmicky přeškálujeme
4. Použijeme *Diskrétní Kosinovu transformaci* (DCT)

Periodogram

Periodogram $Pg_i \in \mathbb{R}^N$ je definován následujícím vztahem

$$Pg_i(k) = \frac{1}{N} B_i(k)^2$$

Melovo škálování

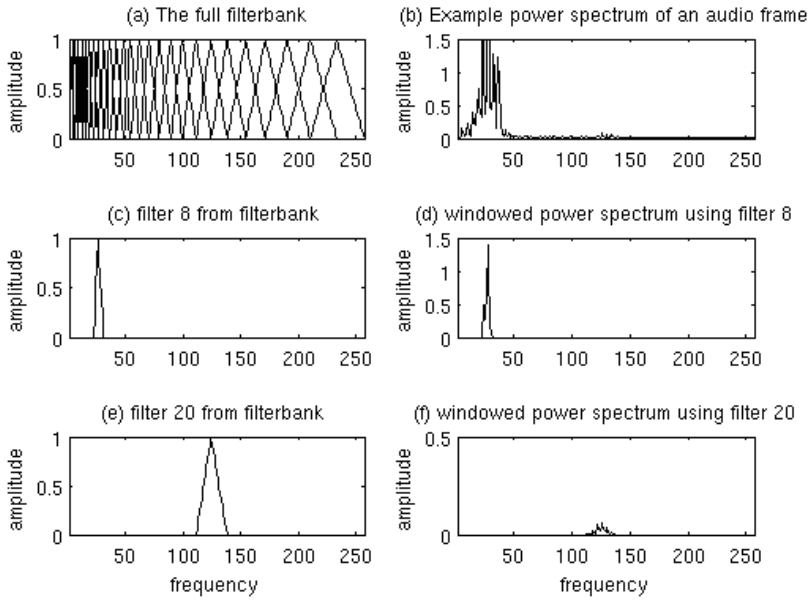
Před dalším krokem si nejprve představíme *Melovo škálování*, Melovo škálování převádí frekvenci tónu do takové podoby, aby co nejlépe vystihoval vnímání lidí. Již jsme si řekli, že lidé vnímáme snadněji změnu výšky v nižších frekvencích. Melovo škálování je definováno následujícím vztahem.

$$Mel(f) = 1127.0148 \log(1 + \frac{f}{700})$$

Melovy filtry

Melovy filtry jsou překrývající se trojúhelníky, které je možné promítnout do periodogramu. V každém filtru potom měříme energii. Princip je znázorněn na obrázku 4.3, kde na levých stranách jsou filtry a na pravých periodogramy po aplikování filtru. Vytvoření takových filtrů je elementární proces, který začíná tím, že si zvolíme počet filtrů a rozsah frekvenčního pásma, který chceme pozorovat. Například stanovíme, že chceme $L = 20$ filtrů mezi frekvencemi 50 Hz až 16000 Hz, dále si rozsah přepočteme podle Melova škálování a dopočítáme si lineárně $L + 2$ bodů m_j mezi rozsahem. Spočteme si lineární krok $step = \frac{Mel(16000) - Mel(50)}{L+1}$ a můžeme definovat $m_j = Mel(50) + (j-1)step$, pro $j = 1, 2, ..., L + 2$. Pole převedeme zpět na Hz inverzní funkcí $Mel^{-1}(mel) = 700(\exp(\frac{mel}{1127.0148}) - 1)$ a vzápětí přepočtem na indexy do periodogramu dostáváme následující formuli.

$$t(j) = \frac{(N+1)Mel(m_j)}{samplerate}$$



Obrázek 4.3: Melovy filtry

Proměnná *samplerate* je hustota dat audionahrávky, my používáme 44 100 Hz. Potom j-tá filtrovací funkce nabývá nulu mimo interval mezi (t_{j-1}, t_{j+1}) a v intervalu trojúhelníkově škáluje s maximem v bodě t_j, její tvar je definujeme následovně.

$$FG_j(k) = \begin{cases} 0 & k < t_{j-1} \\ \frac{k-t_{j-1}}{t_j - t_{j-1}} & t_{j-1} \leq k \leq t_j \\ \frac{k-t_j}{t_{j+1}-t_j} & t_j < k \leq t_{j+1} \\ 0 & k > t_{j+1} \end{cases}$$

Máme dostatečné prostředky na to, abychom vyjádřili energii $SE_i(j)$ pro každý filtr $j = 1, ..., L$, pro výpočet použijeme periodogram násobený filtrační funkcí.

$$SE_i(j) = \sum_{k=0}^{N} Pg_i(k) FG_j(k)$$

Poslední dva kroky výpočtu vyjádříme závěrečným vztahem pro j-tý koeficient.

$$MFCC_i(j) = \sum_{k=1}^{L} \log(SE_i(j)) \cos[j(k - \frac{1}{2})\frac{\pi}{L}]$$

Mohla by nás napadnout otázka, proč pouštíme na energii logaritmus, když bychom očekávali třetí odmocninu. Logaritmus umožňuje *cepstral mean sub-*

traction, která funguje díky větám o logaritmech, pro popis odkazuji na zdroj [7].

4.4 Chromatický vector

Chromatický vektor je velice často používaný deskriptor v aplikacích pracující s hudbou [5, 6]. Vektor má 12 prvků, tedy přesně kolik máme různých not v jedné oktávě včetně *půltónů*. Chromatický vektor nám ke každé notě hodnotu, která určuje její výskyt. Představme si, že máme dvanáct krabiček a každou popíšeme jednou notou. Poté roztřídíme magnitudy spektra B_i podle frekvence do odpovídající krabičky a nakonec bychom v každé krabičce spočetli aritmetický průměr.

$$\{C, C_\#, D, D_\#, E, F, F_\#, G, G_\#, A, A_\#, B\}$$

Formálně definujeme chromatický vektor takto. Nechť máme $k = 1, ..., 12$, množinu indexů do spektra odpovídající k-té notě S_k a její velikost $M_k = |S_k|$, potom chromatický vektor je zadán následujícím vztahem.

$$CV_i(k) = \sum_{j \in S_k} \frac{B_i(j)}{M_k}$$

Indexy do spektra si můžeme přepočítat na frekvence $fr_t = \left\lfloor \frac{(1+t)samplerate}{2N} \right\rfloor$ frekvence převedeme na noty vztahem.

$$Note_t = round(12 \log_2(\frac{fr_t}{27.5}))$$

Potom definujeme množinu indexů vztahem.

$$S_k = \{t : Note_t \equiv k \pmod{12}\}$$

.

4.5 Odhad tempa

Tempo se vyjadřuje různým způsobem, můžeme se setkat s počtem taktů za minutu nebo slovní popis italskými slovy (*Moderato*, *Allegretto*, *Allegro* apod.), které označují rozsah. My budeme používat třetí způsob, které tempo uvádí v jednotce BPM (*beats per minute*) tedy počet úderů za minutu. Úderem se zde rozumí *čtvrťová nota*. Jednotka BPM vyjadřuje kolik čtvrťových dob bude mít jedna minuta.

Pro odhad tempa provádíme následujícím postupem.

1. Signál máme navzorkovaný na malé rámce (25 ms) a pro ně spočtené MFCC vektory. Vytvoříme si velké rámce (6 s) a pro ně vypočteme *Self-Similarity Matrix* (SSM) pomocí *Eukleidovské vzdálenosti.* Získáváme čtvercovou matici o velikosti $\frac{6\ s}{25\ ms} = 240$.

$$SSM(i,j) = \sqrt{\sum_{k=1}^{12}(MFCC_i(k) - MFCC_j(k))^2} \quad \text{pro} \;\; i,j \in \langle 1, 240\rangle \cap \mathbb{Z}$$

   Tato matice se vyznačuje tím, že má na hlavní diagonále nuly $SSM(i,i) = 0$ a je symetrická $SSM(i,j) = SSM(j,i)$.

2. V pravém horním trojúhelníku matice nad hlavní diagonálou $SSM(i,j) \;\; i < j$ mají diagonální vektory souřadnice ve tvaru $(i, i+k)$ pro $k = 1, ..., 239$. Spočteme aritmetický průměry těchto vektorů a označíme je jako signál T.

3. Okolí vektoru T si dodefinujeme nuly $T(k) = 0, \quad k \leq 0 \;\vee\; k \;\geq 240$. Spočteme odhad druhé derivace signálu T, podle formule.

$$D_1(k) = \sum_{h=-3}^{3} T(k+h), \quad k = 1, ..., 239$$

   Dodefinujeme $D_1(k) = 0, \;\; k \leq 0 \;\vee\; k \;\geq 240$.

$$D_2(k) = \sum_{h=-3}^{3} D_1(k+h), \quad k = 1, ..., 239$$

4. Ve vektoru D_2 si nalezneme lokální maxima a ukládáme jejich relativní indexy vůči prvnímu z nich M. Známe hodnotu *posun*, která vyjadřuje index $M(1)$ v poli D_2.

5. Hledáme dvojici $a, b, \quad a < b$ lokálních maxim, která nejlépe splňuje následující dvě kritéria. Prvním kritérium zní $M(b)$ dělí $M(a)$ s nejnižším možným zbytkem. Druhé kritérium zní, že součet těchto maxim je co největší $D_2(posun + M(a)) + D_2(posun + M(b))$.

6. Výsledné tempo se počítá následující formulí, ke a je menší prvek z dvojice, která nejlépe splnila kritéria.

$$Tempo = \frac{60}{0.025 M(a)}$$

Kapitola 5

Vytvoření datasetu

Doteď jsme se věnovali analýze zvuku, kterou potřebujeme k tomu, abychom z nahrávek mohli extrahovat nějaká data, která budou nahrávky reprezentovat. V této části si vytvoříme *dataset*, což je množina dat zapsaná v maticovém formátu. V datasetu bude u každé skladby taneční styl jako název třídy (*label*) a seznam extrahovaných hodnot. Pro reprezentaci datasetu použijeme formát CSV.

Feature extraction

Implementovat vlastní knihovnu pro extrakci základních mpeg-7 deskriptorů nemá příliš smysl, už jenom z důvodu, že se jedná o celosvětově rozšířený standard. Dokončených implementací se na internetu vyskytuje hned několik. Takové komplexní řešení, je například open-source knihovna *essentia*, která shromažďuje a obaluje veliké množství deskriptorů. Essentia klade důraz na robustnost a optimalizaci výpočtu, ovšem využívá veliké množství knihoven od třetích stran a je komplikované ji nasadit.

Vhodné řešení pro náš projekt je použít *pyAudioAnalysis*, je to implementace v Pythonu přímo od T. Giannakopoulose, jednoho z autorů knihy o analýze dat [6]. V našem případě nám jde o jednorázové použití, takže nás nebude příliš trápit rychlost knihovny. Navíc implementaci ve vyšším programovacím jazyku jako je Python si můžeme snadno ověřit, zda odpovídá matematickým popisům. Kromě rychlosti má knihovna nevýhodu, že zvládá pouze jeden nekomprimovaný formát WAVE.

Sjednocení formátů

Testovací archiv, který máme k dispozici od pana Ing. Jana Trávníčka, obsahuje 664 skladeb rozdělené do devíti tříd podle tanečních stylů. Nahrávky jsou ve formátu MP3 nebo OGG. Hustota záznamu (samplerate) je vetšinou 44 100 Hz v některých případech 22 050 Hz, většinou se jedná o stereo. Převedeme proto všechny data na společní formát WAVE.

Použijeme nástroj ffmpeg pro převody audiovizuálních dat. Zároveň při převodu si sjednotíme i samplerate na 44,1 KHz.

```
$ ffmpeg -v 0 -i input.mp3 -ar 44100 output.wav
```

Seznam atributů

Tabulka 5.1: Seznam atributů datasetu

id	název atributu
1	Genre
2	BPM
3	confidence
4	ZRC
5	Energy
6	EntropyOfEnergy
7	SpectralCetroid
8	SpectralSpread
9	SpectralEntropy
10	SpectralRolloff
11-23	MFC vector
24-36	Chroma vector
37	ChromaDeviation

5.1 Statistiky datasetu

Data máme celkem poměrově dobře vyvážená, od všech tříd máme cca 90 zástupců. Jediné minoritní skupiny jsou salsa a polka, ale to jsou pouze dvě třídy z devíti, vliv na chybu nebude příliš velký.

Tabulka 5.2: Statistika datasetu.

třída	četnost
ChaCha	99
Jive	85
Polka	14
Rumba	94
Salsa	7
Samba	77
Tango	100
Valcik	81
Waltz	107

Zhodnocení kvality odhadu BPM

Tempo je přímo rys tance a pouze podle tempa bychom byli schopni často rozhodnout. To nás vede k tomu dát takovému atributu větší váhu. Nejprve si ale však vyjádříme chybu odhadu. Metoda odhadu tempa, kterou jsme použili, má velice omezený počet výsledných hodnot. Například v množině hodnot máme hodnotu 120 a další vyšší hodnota je 133. Což by ovšem mohlo znevýhodnit tanec cha cha, který má rozsah 120 až 130 BPM. Nahrávka s tempem 128 by se spíše uchýlila k hodnotě 133. Stanovili jsme kritérium pro správný odhad tempa, které říká, že odhad je správný, právě když se vejde do tolerovaného rozsahu. Tolerovaný rozsah je tvořen čísly z množiny možných hodnot odhadu a je zvolen tak, aby se do něj vešel charakteristický rozsah pro tanec.

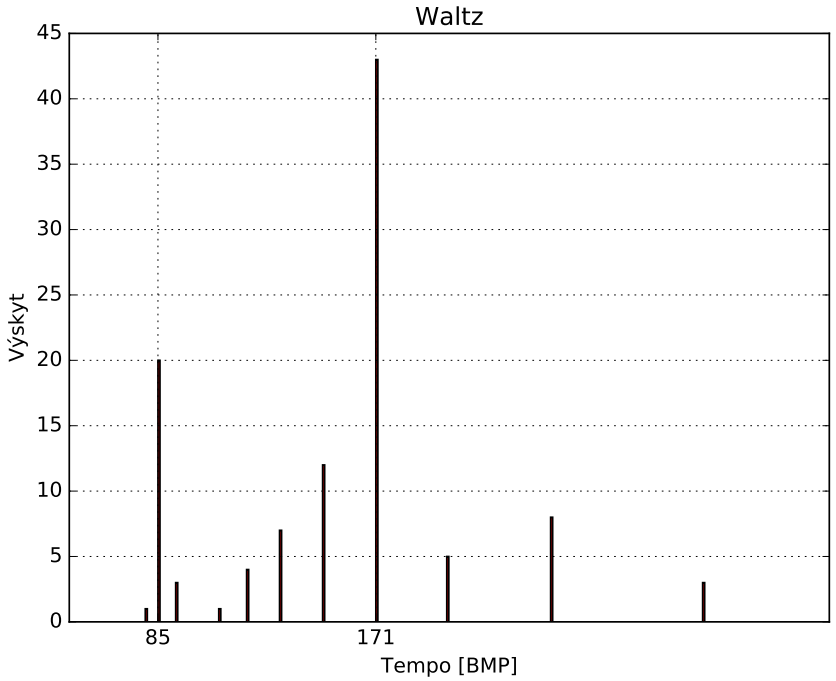
Tabulka 5.3: Chyba odhadu tempa

třída	tolerovaný rozsah	průměr	chyba
ChaCha	120 až 133	141	22.22 %
Jive	150 až 199	120	60.00 %
Polka	99 až 120	117	7.14 %
Rumba	99 až 110	165	81.91 %
Salsa	150 až 240	150	14.29 %
Samba	99 až 109	110	25.97 %
Tango	120 až 133	128	15.00 %
Valcik	171 až 199	168	8.64 %
Waltz	80 až 92	154	77.57 %

Chyba je spočítána jako počet špatných odhadů vůči celku. Tabulka 5.3 znázorňuje dílčí chyby v jednotlivých třídách. Jak je vidět z tabulky, odhad pro některé třídy je téměř nefunkční.

Celková chyba vychází 41,7 %, mohli bychom si dojít k závěru, že odhad je velice nepřesný a atribut nám zavádí do klasifikace pouze šum do datasetu. Ovšem nemusí to tak zcela nutně být, nepřesnost odhadu neimplikuje zavádění šumu. Protože nemáme dostatek časové dotace pro hledání algoritmu na přesnější odhad tempa, zkusíme si zanalyzovat co nám atribut dává za informace ke klasifikaci. Budeme podrobně zkoumat chyby, podíváme se u všech tříd, kam se hodnoty zobrazí a zjistíme zda vytvářejí nějaké nové shluky nebo jsou nepředvídatelně rozptýlené.

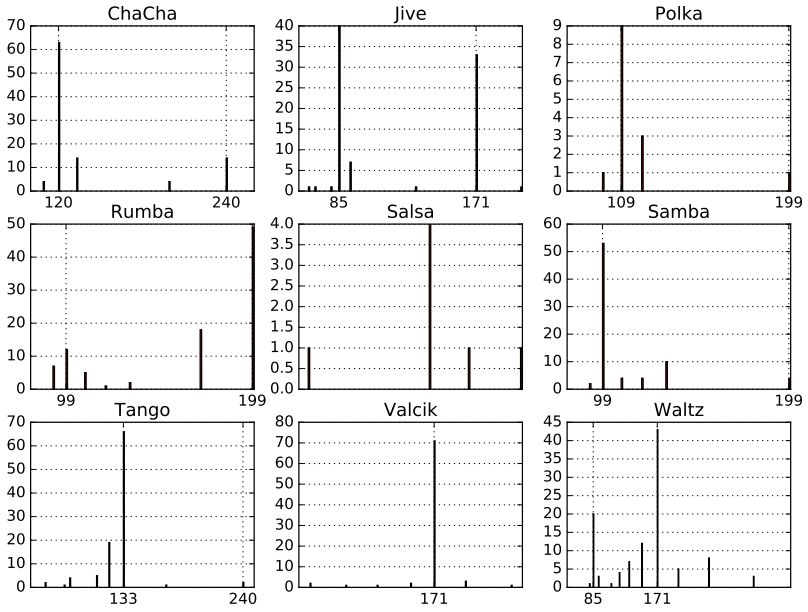
Jeden z nejhorších výsledků měl tanec waltz, a proto ho budeme dále analyzovat. U všech našich výsledků odhadu tempa pro waltz si napočítáme četnost, abychom zjistili jak chyby vznikají. Z grafu na obrázku 5.1 je vidět, že nejčastější dva výsledky jsou 85 a 171. Hodnota 85 je správný odhad a hodnota 171 je nejbližší možná hodnota dvojnásobku správné hodnoty. Otázka, proč metoda často dává dvojnásobek, se dá vysvětlit tím, že mezi údery tempa se vyskytuje nějaký tón, který úder připomíná.



Obrázek 5.1: Rozložení odhadů pro waltz

Podívejme se na ostatní tance 5.2. Distribuce u odhadů pro cha chu je méně zašuměná, obsahuje tři frekventované hodnoty 120, 133 a 240. Zde je ještě zřetelnější problém s dvojnásobkem. U tance jive se nám ukazuje, že častá chyba je i poloviční hodnota. U polky máme menší četnost dat, ale i tam distribuce vytváří dva shluky. Od salsy máme pouze 7 zástupců a navíc má salsa veliký rozsah, proto z ní moc nevyčteme. Rozložení ve třídách samba, tango a valčík má po jednom shluku.

Po analýze chyby můžeme shrnout, že nejčastěji distribuce odhadu vytváří dva shluky. Minoritní z těchto shluků může být výrazně menší. Příliš malé shluky se klasifikátor pravděpodobně nedoučí, ale tím, že jsou malé, budou mít i malý vliv na klasifikační přesnost. Třídy jive, valčík a waltz se překrývají na hodnotě 171. U dvojice waltz a valčík je to škoda, protože tempo je jediný jejich charakteristický rozdíl v tabulce 2.1.



Obrázek 5.2: rozložení odhadů pro všechny třídy

Kapitola 6

Klasifikace

V této kapitole budeme hledat nejlepší *klasifikační model*. Zvážíme a vyzkoušíme si vhodné předzpracování dat. Budeme hledat nejlepší klasifikátor a jeho nejlepší konfiguraci. Stavový prostor při hledání ideálního modelu je enormní a není možné ho celý prohledat. Máme k dispozici 36 atributů, tedy $2^{36} - 1$ možností jakou podmnožinu vybrat. Každý atribut můžeme přeškálovat na nespočetné množství způsobů. Otestuje klasifikátory *k-NN*, *rozhodovací strom* a *Bayesův klasifikátor*. Po každý klasifikátor vytvoříme model, který budeme vylepšovat. Hrubou silou budeme zkoušet pouze malé části stavového prostoru, které půjdou prohledat v rámci desítek minut. Na závěr srovnáme výsledky.

Poznámky

- Pro vytvoření klasifikačního modelu jsme využili nástroj *RapidMiner* namísto *Matlabu*. A to z důvodu, že s RapidMinerem jsem měl předchozí zkušenosti z předmětu VZD. Takovou změnu jsme nepovažovali za příliš zásadní, abychom kvůli ni nechávali změnit odsouhlasené zadání práce. Klasifikační modely vytvořené v RapidMineru a Matlabu jsou vzájemně převoditelné.

- Klasifikace, kde klasifikátor doporučuje seznam tříd, se nazývá *multi-label classification* a umožňují ji pouze některé typy klasifikátorů. Těmi jsou například k-NN nebo *boosting*, což je pokročila technika, která slučuje několik klasifikačních modelů dohromady. U ostatních klasifikátorů, které tuto možnost nenabízí budeme říkat, že výstupem jejich klasifikace je jednoprvkový seznam a tedy je můžeme zahrnout pro splnění cíle práce.

Křížová validace

Pro měření klasifikační přesnosti použijeme *křížovou validaci* (zk. X validace). X validace provádí více měření a vrací průměrnou úspěšnost. Tento algoritmus

je závislý na parametru p. Dataset rozdělí na p stejně velkých částí a jednu z nich (i-tou) použije pro naučení modelu. Ostatních $k-1$ částí použije pro testování. Proces se opakuje pro $i = 1, 2, \ldots, p$.

Při rozdělování dat do skupin, tvoříme skupiny se stejným procentuálním zastoupením klasifikačních tříd. Tím si zajistíme, aby vždy všechny skupiny byly obsaženy v části určené pro učení modelu, pokud je v nejmenší třídě alespoň p vzorků.

6.1 Klasifikátor rozhodovací strom

Jako první klasifikátor vyzkoušíme rozhodovací strom (decision tree). Důvody proč volíme rozhodovací strom jsou, že algoritmus pro vytvoření rozhodovacího stromu zahrnuje odhad významnosti atributů. Výsledný strom má nejvyšších patrech (roste dolů) nejvýznamnější parametry. Není sice pravda, že nejvýznamnější parametry pro rozhodovací strom musí být nejvýznamnější pro jiné klasifikátory, ale je to pravděpodobné. Pro rozhodovací strom nemusíme hledat podmnožinu atributů a je nezávislí na lineárním přeškálováním parametrů. Rozhodovací nemá příliš mnoho dimenzí, ve kterých by se dal konfigurovat.

princip

Rozhodovací strom má v každém vnitřním uzlu identifikátor parametru a intervaly určující jeho potomky. Každý list stromu je asociován právě jedné třídě. Na základě trénovacích dat `data` a *kritériu výběru rozčlenění* `criterion` vytvoříme strom rekurzivním algoritmem.

```
def BuildTree( node, data, criterion ):
        if data.hasOnlyOneClass :
            node.class = data.class
            return
        attrId := FindAttribToSplit( data, criterion )
        children := SplitDataBy( data, attrId )
        for child in children:
            BuildTree( child.node, child.data, criterion )
```

U tohoto klasifikátoru můžeme konfigurovat kritérium výběru rozčlenění a omezit maximální hloubka. Nejobvyklejší kritéria jsou *informační zisk* a *gain ration*. V rapidmineru máme dokonce 4 kritéria informační zisk, gain ratio, *gini index* a *klasifikační přesnost*.

- informační zisk: K rozdělení bude vybrán parametr s nejnižší entropií. Toto kriterium diskredituje výběr atributů, které mají mnoho hodnot.

$$SplitEntropy(S, A) = -\sum_{V \in Values(A)} \frac{|S_V|}{|S|} \log[2] \frac{|S_V|}{|S|}$$

Pro atribut A, množinu dat S a $S_V = \{x : x \in S, x_A = V\ \}$.

- klasifikační přesnost: Je vybrán parametr, který maximalizuje přesnost pro celý strom [8]. Toto kritérium není příliš obvyklé a nenašli jsme jeho formální popis.
- gini index: Měří míru nejednoznačnosti v S pomocí Giniho koeficientu, vybere atribut, který nejlépe sníží míru nejednoznačnosti. Atribut s nejnižší hodnotou bude rozdělen.

$$Gini(A) = \sum_{V \in Values(A)} \frac{|S_V|}{|S|} \prod_{B \in Values(S_V)} \frac{|S_B|}{|S_V|}$$

- gain ratio: Je varianta informačního zisku, která penalizuje atributy s větším s velkým počtem hodnot.

$$GainRatio(S, A) = \frac{Gain(S, A}{SplitEntropy(S, A)}$$

$$Gain(S, A) = H(S) - \sum_{VinValues(A)} \frac{|S_V|}{|S|} H(S_V)$$

Oba konfigurační parametry rozhodovacího stromu můžeme hledat hrubou silou. Jedna X validace pro rozhodovací strom trvá přibližně 1,2 sekundy (průměr ze 100 měření na procesoru i5 o taktu 2 364 MHz). Zkusíme-li na hledání maximální hloubky 60 pokusů, výpočet bude trvat přibližně 5 minut. Budeme hledat maximální hloubku v rozsahu $1, 2, ..., 60$.

Malá hloubka je pro rozhodovací strom limitující faktor, proto s přidávající hloubkou roste klasifikační přesnost. Přesnost u všech kritérii nakonec konverguje k hodnotě 65 % při hloubce 5 a víc. Klasifikační přesnost u kritéria gain ratio roste výrazně pomaleji. Kritérium information gain dosahuje v průměru nejlepších výsledků a jeho maximum je 71,8 %.

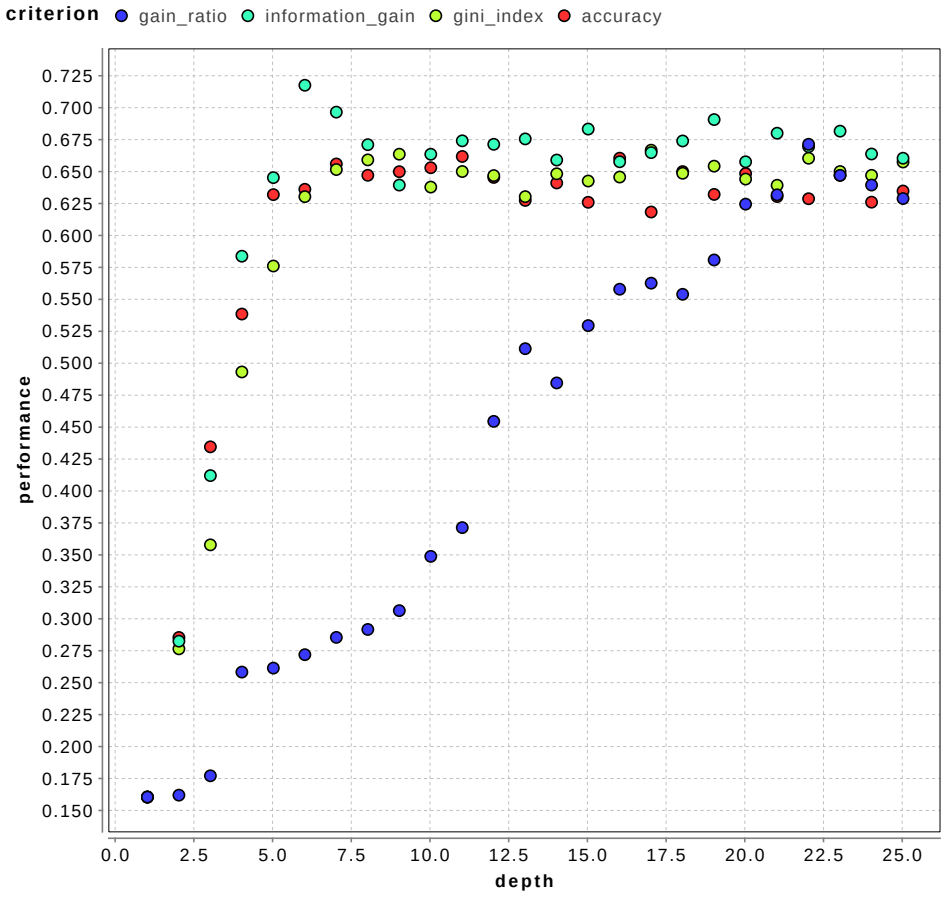
Výsledný rozhodovací strom je příliš obsáhlý na to, aby ho bylo možné zobrazit na stránku. Ovšem použil 21 parametrů z 36 a to jsou:

`spectral centroid, BPM, spectral rolloff, energy, confidence, ZRC, entropie energie, MFC{2,3,4,6,7,8,9,10,13} a CV{1,2,5,8,13}.`

6.2 Klasifikátor k-NN

Klasifikátor k-NN neboli K nejbližších sousedů si v trénovací fázi uloží trénovací data, která jsou obvykle ve tvaru název třídy a číselný vektor. Pro klasifikaci nového prvku x si najde k nejbližších sousedů a jejich statisticky nejpočetnější třída je přiřazena x.

Naivní implementace by měla složitost jedné klasifikace $\theta(n)$, proto se jako optimalizace používají datové struktury quad tree nebo KD tree.



Obrázek 6.1: Klasifikační přesnost pro rozhodovací strom

Metrika

Pro výpočet vzdálenosti dvou vzorků datasetu $\mathbb{S} \subset \mathbb{R}^n$ můžeme použít libovolnou metriku vzdálenosti $d : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$, která splňuje následující podmínky.

$$\begin{aligned} &d(x,y) \geq 0 \quad \text{(nezápornost)} \\ &d(x,y) = 0 \ \text{ iff } \ x = y \\ &d(x,y) = d(y,x) \quad \text{(symetrie)} \\ &d(x,y) \leq d(x,z) + d(z,y) \quad \text{(troj. nerovnost)} \end{aligned} \tag{6.1}$$

Nejčastěji používaná je eukleidovská vzdálenost, která v trojrozměrném prostoru se standardní bází určuje vzdálenost vzdušnou čarou.

- Eukleidovská vzdálenost

$$d(x, y) = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}$$

- Manhattanská vzdálenost

$$d(x, y) = \sum_{j=1}^{n} |x_j - y_j|$$

- Čebyševova vzdálenost

$$d(x, y) = \max_{j=1}^{n} |x_j - y_j|$$

- Canberrova vzdálenost

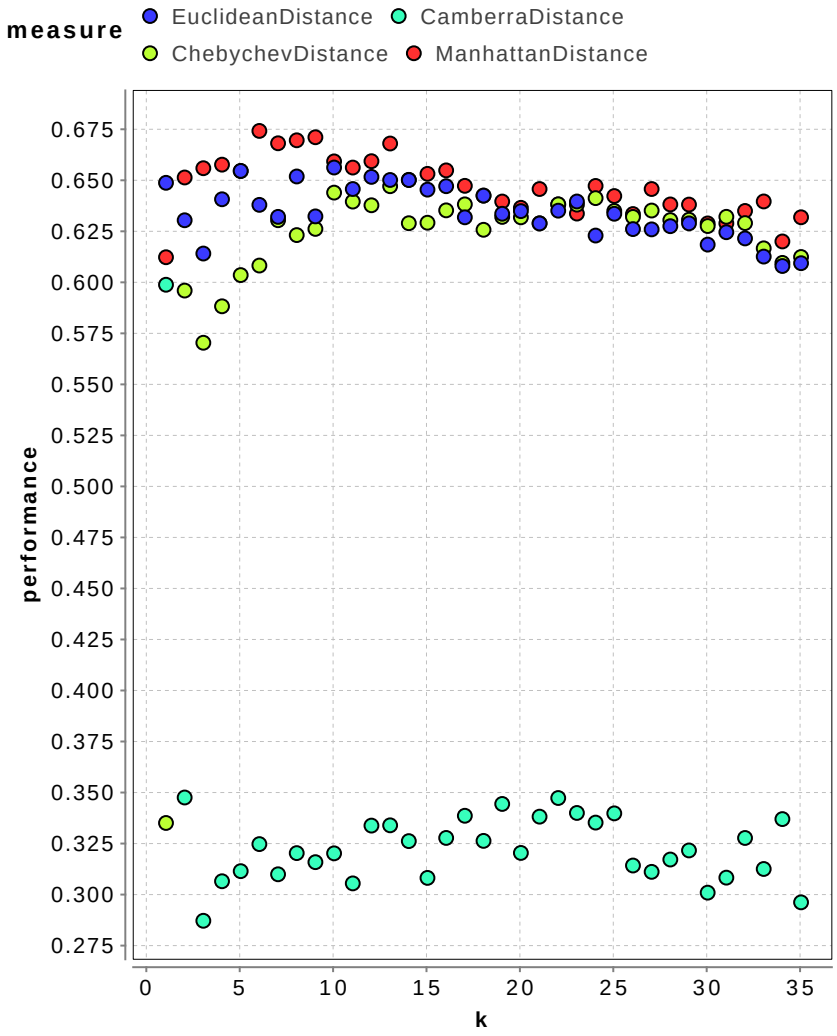
$$d(x, y) = \sum_{j=1}^{n} \frac{|x_j - y_j|}{|x_j| + |y_j|}$$

Hledání hodnoty K

Parametr k je možné nastavit od 1 až po velikost množiny trénovacích dat, což by nám vracelo pouze nejčastější hodnotu v trénovací množině. Pro příliš malé k je klasifikace silně ovlivněna šumem v datech. Příliš velké k má nevýhodu, že utlumí vliv malých shluků dat. Z tabulky 5.2 si můžeme zjistit rozložení trénovací množiny. Hodnoty dělit parametrem p z X validace, který jsme nastavili na 10. Dostáváme informaci, že minimální průměrný počet výskytů třídy v trénovacích datech je 1, maximum je 11 a medián je 9.

V sekci o zhodnocení odhadu BPM v předchozí kapitole 5.1 jsme došli k závěru, že většina atributů tvoří v tomto parametrů dva shluky. A z multigrafu 5.2 vidíme že významné shluky mají přibližně 2, 3 a 4, ale i 9 zástupců po vydělení p. Z toho můžeme predikovat, že ideální k by mohlo být okolo hodnoty 4 až 8, aby malý shluk nebyl překryt nejbližším větším shlukem. Ale úvaha zvažuje pouze jeden parametr, proto raději vyzkoušíme větší rozsah 1 až 35.

Výsledky měření klasifikační přesnosti 6.2 ukazují, že Canberova vzdálenost se na tento případ příliš nehodí, naopak nejvhodnější metrikou je Manhattanská vzdálenost. Nejlepší hodnota pro k je v rozsahu 6 až 9 a oddalováním od tohoto intervalu klesá klasifikační přesnost. Nejlepší výsledek je 67,5 % při $k = 6$. Toto k si ponecháme a bude se zažit optimalizovat jinými způsoby.



Obrázek 6.2: Hledání k a metriky.

Hledání nejvhodnější podmnožiny atributů

Některé atributy by mohli mít negativní vliv na klasifikaci. Hledání ideální podmnožiny je ovšem výpočetně příliš složité, jak již bylo zmíněno. Můžeme vyzkoušet některé konkrétní n-tice. Počet n-tic pro 36 prvků je $\binom{36}{n}$. Vezmeme-li v úvahu symetrii Pascalova trojúhelníku, můžeme prohledat menší n-tice a také všechny doplňky k nim. Zvládneme najít nejlepší jeden prvek, dvojici, trojici, 35-tici, 34-tici a 33-tici.

Nejlepší výsledek dává následující trojice s klasifikační přesností 69,15 %. (`BPM`, `chorma vector 12`, `spectral centroid`)

Experiment vybrat atributy, které použil rozhodovací strom vedl k výsledku 65,4 %.

Váhy atributů

U k-NN má atribut s největším rozptylem největší váhu pro klasifikování. Abychom dali všem atributům stejnou váhu, museli bychom je všechny normalizovat na stejný rozsah. Problém najít ideální podmnožinu se dá redukovat na hledání vah atributů (prvkům, které chceme vyřadit přidělíme nulovou váhu). Použili jsme genetický algoritmus, který prohledá alespoň část stavového prostoru. Výsledná klasifikační přesnost je 73,50 %.

Tabulka 6.1: Výsledky klasifikace: Ve sloupcích jsou skutečné vzorků a v řádcích jsou znázorněny třídy, které klasifikátor predikoval. Například v prvním sloupci vidíme, že 9 nahrávek cha cha bylo chybně klasifikovaných jako tango. CP (class precision) říká kolik označení do té třídy bylo správně. CR (class recall) říká kolik bylo správně klasifikováno z dané třídy.

-	CH	VA	SL	PO	JI	RU	SB	TG	WT	CP[%]
CH	83	0	0	1	0	1	9	12	3	76
VA	0	64	0	0	2	14	0	1	23	62
SL	0	0	1	0	0	0	0	0	0	100
PO	1	0	0	9	0	1	2	4	1	50
JI	0	1	1	0	77	2	2	4	4	85
RU	4	4	0	2	0	65	2	1	7	76
SB	2	0	1	0	1	4	57	0	0	88
TG	9	3	0	2	3	3	5	73	10	68
WT	0	9	4	0	2	4	0	5	59	71
CR[%]	84	79	14	64	91	69	74	73	55	

6.2.1 Komentář k výsledkům

Nejvíce chyb vzniklo chybným označením elementů z třídy waltz jako valčík (23 chyb). Logicky se tato chyba projevovala i na druhou stranu, některé elementy valčíku byly označeny jako waltz (9 chyb). To je způsobeno tím, že se nám nepovedlo odhadnou tempo pro waltz správně a waltz má stejné charakteristiky s valčíkem vyjma tempa. Další zdroj chyb je špatná klasifikace elementů rumby jako valčík (14 chyb). Na druhou stranu se takto chyba příliš neprojevovala (4 chyby). To značí, že rumba měla minoritní shluk u valčíku, ale $k = 6$ bylo příliš velké, aby byla predikce správná. Poslední dvojice, která tvoří hodně chyb je tango a cha cha, kde 12 elementů tanga bylo klasifikováno jako cha cha a naopak vzniklo 9 chyb. Ze statistik se tango a chacha příliš neliší, rozhodnout by se dalo podle taktu nebo rytmických vzorků.

6.3 Bayesův klasifikátor

Bayesův klasifikátor funguje na principu Bayesovy věty.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(A \cap B) = P(B|A)P(A)$$

Výraz $P(A)$ označuje pravděpodobnost, že nastane jev A. Podmíněná pravděpodobnost $P(A|B)$ je pravděpodobnost, že nastane jev A, pokud nastal jev B. Výraz $P(A \cap B)$ je pravděpodobnost, že nastanou jevy A a B zároveň.

Definice (Nezávislost)**.** *Jevy A a B jsou nezávislé, právě když platí následující rovnost.*

$$P(A \cap B) = P(A)P(B)$$

Naivní předpoklad

Bayesův klasifikátor předpokládá, že pro jevy $X_1, X_2, ..., X_n$ vždy platí.

$$P(X_1, X_2, ..., X_n|A) = P(X_1|A)P(X_2|A)...P(X_n|A)$$

Tedy předpokládá, že všechny jevy jsou si vzájemně nezávislé.

Princip

Klasifikátor si statisticky vyjádří pravděpodobnosti jednotlivých jevů za podmínky klasifikační třídy h. Poté vyhledává nejpravděpodobnější kombinaci jevů pro predikci třídy h. Pro podrobnější popis doporučuji zdroj [9].

6.3.1 Výsledek měření

Klasifikační přesnost z X validace byla 59,18 %.

Tabulka 6.2: Srovnání klasifikátorů

Metoda	Rozhodovací strom	k-NN	Bayesův klasifikátor
Nejlepší výsledek	71,8 %	73,50 %	59,18 %

Závěr

Představili jsme si současný stav v oblasti klasifikace hudby. Zjistili jsme, že existuje několik žánrových klasifikátorů, ale nepodařilo se nám nalézt žádné hotové řešení pro rozpoznávání tanečních stylů. Seznámili jsme se s pokročilou analýzou zvuku a hudby, kterou jsme následně použili na testovací archiv s taneční hudbou. Z každé nahrávky jsme vytěžili všech 34 základních a spektrálních deskriptorů popsaných standardem MPEG-7 a dva z odhadu tempa, pomocí knihovny pyAudioAnalysis.

Na vzniklém datasetu jsme provedli vytvořili několik klasifikačních modelů, které jsme postupně vylepšovali. Výsledky nejlepších výkonů jednotlivých klasifikátorů jsme zobrazili do tabulky 6.2. Nejlepší klasifikační přesnosti dosáhl klasifikátor k-NN s výsledkem 73.5 %. Pro měření klasifikační přesnosti jsme použili křížovou validaci.

Výhledy do budoucna

Prací se rozhodně má smysl zabývat do budoucna. Pro zlepšení klasifikační přesnosti, bych hledal nové metody pro odhad tempa. Svojí pozornost by si zasloužily další extrakce parametrů, například rytmické vzorky. Dále bych zkusil jiné klasifikátory například samoorganizující se mapy.

Pokud se povede dosáhnout alespoň 95 % přesnosti, vyplatilo by se zrealizovat mobilní aplikaci pro začínající tanečníky, která z mikrofonového vstupu bude schopná rozeznat tanec z několika prvních sekund skladby.

Literatura

[1] T. Lidy, A. Rauber, *Machine Learning Techniques for Multimedia*, Springer-Verlag Berlin Heidelberg, ISBN: 978-3-540-75170-0, 2008.

[2] M.Haggblade, Y. Hong, K. Kao, Music Genre Classification [online], Stanford, 2011, cit. 29.11.2015 dostupné z http://cs229.stanford.edu/proj2011/HaggbladeHongKao-MusicGenreClassification.pdf

[3] TZANETAKIS, George, ESSL, Georg and COOK, Perry. Automatic Musical Genre Classification Of Audio Signals [online]. Princeton, no date. cit. 2.12.2015 dostupné z http://ismir2001.ismir.net/pdf/tzanetakis.pdf

[4] HAGGBLADE, Michael, Yang HONG a Kenny KAO. Music Genre Classification [online]. [cit. 2015-12-5]. Dostupné z: http://www.cs.cmu.edu/ yh/files/GCfA.pdf

[5] KIM, Hyoung-Gook, MOREAU, Nicolas and SIKORA, Thomas. MPEG-7 audio and beyond: audio content indexing and retrieval. Chichester, West Sussex, England : J. Wiley, 2005.

[6] GIANNAKOPOULOS, Theodoros and PIKRAKIS, Aggelos. Introduction to audio analysis: a MATLAB approach. Waltham, USA : Elsevier Ltd., 2014.

[7] LYONS, James. Mel Frequency Cepstral Coefficient (MFCC) tutorial. Practical graphy [online]. 2013. [cit 2016-05-10]. Retrieved from: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[8] Decision tree. RapidMiner documentation [online]. [cit. 2016-05-16]. Dostupné z: http://docs.rapidminer.com/studio/operators/modeling/ classification_and_regression/tree_induction/decision_tree.html

[9] Naive Bayes classifier [online]. [cit. 2016-05-12]. Dostupné z: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[10] ZIKMUNDOVÁ, Romana. HISTORICKÝ VÝVOJ A CHARAKTERISTIKA STANDARDNÍCH [online]. Plzeň, 2013 [cit. 2016-05-16]. Dostupné z: https://otik.uk.zcu.cz/bitstream/handle/11025/7083/ Bakalarska prace_R.Zikmundova.pdf. Bakalářská práce. Západočeská univerzita v Plzni. Vedoucí práce Mgr. Petra Kalistová.

[11] Rumba. Supertanec.cz [online]. [cit. 2016-05-16]. Dostupné z: http://www.supertanec.cz/doku.php?id=rumba

[12] VÍCHOVÁ, Ilona. 3 základní tance, které byste měli znát!. Žena [online]. 2013, 2013, 4 [cit. 2016-05-16]. Dostupné z: https://goo.gl/7JQoxL

Příloha **A**

Seznam použitých zkratek

AV audiovisual - audiovizuální

BPM Beats per minute - počet úderů za minutu

CSV Coma Separeted Values - formát pro ukládaní dat maticového tvaru

DCT Discrete Cosine Transform - diskretní kosinova transformace

DFT Discrete Fourier Transform - diskrétní Fourierova transformace

FFT Fast Fourier Transform - rychlá Fourierova transformace

k-NN k-Nearest Neighbors - K nejbližších sousedů

MFCC Mel-Frequency Cepstrum Coefficients - cepstralní koeficienty Melových frekvencí

MIR Music Information Retrieval, výzkumná skupina na TUV

MP3 MPEG-1 Audio Layer 3 populární audio formát se ztrátovou kompresí.

SSM Self-Similarity Matrix - matice vzájemné podobnosti

WAVE Waveform audio file format - reprezentace tlakové vlny

Příloha B

Seznam použitých techonologií

Python a balíčky

Python je moderní a v poslední době velice oblíbený skriptovací jazyk, který je vhodný k rychlému vytváření prototypů. Jeho obliba je především dána jednoduchou syntaxí a přehledným kódem. Jedná se o vysokoúrovňový netypový jazyk s prvkami funkcionálního programování. Pro python vzniklo veliké množství balíčků, nejznámější z nich jsou například numpy a scipy, které jsou určené pro matematické a věděcké účely.

FFMPEG

FFMPEG je víceplatformní nástroj pro nahrávání, převádění a streamování audia a videa. Dostupný z `https://ffmpeg.org/`.

PyAudioAnalysis

PyAudioAnalysis knihovna pro analýzu zvukových nahrávek, založena na standardech MPEG-7. Tuto implementaci v jazyce python napsal T. Giannakopoulos, jeden z autorů knihy [6]. Za pomocí balíčku scipy a numpy.

RapidMiner

RapidMiner je komplexní nástroj pro datovou vědu, umožňuje načítat a vizualizovat data, předzpracování dat, klasifikaci, shlukování, optimalizace parametrů a mnoho dalšího.

Příloha C

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `data` ..... Adresář s podpůrnými materiály `.1 src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF