K-nearest neighbors (KNN)

KNN nu construieşte alt model decât acela de a memora întregul set de date de antrenare, deci nu este nevoie de o etapă de învăţare. Deoarece setul de date este memorat, aceste date trebuie să fie consistente. În practică, se recomandă ca setul de date sa fie pre-procesat. Actualizarea modelului cu date noi se poate face foarte uşor, fără să fie nevoie de procesări complexe.

Predicţia pentru o nouă instanţă (X) se bazează pe căutarea în întregul set de date a celor mai asemănătoare K instanţe (a celor mai apropiaţi K vecini). În cazul regresiei rezultatul va fi media valorilor vecinilor, iar în cazul clasificării rezultatul va fi categoria cu cel mai mare număr de vecini.

Pentru a determina care sunt cele mai apropiate K instanţe cu instanţa ce trebuie clasificată trebuie să se folosească o măsură de distanţă. Pentru valorile reale ale variabilelor de intrare, cea mai cunoscută măsură de distanţă este distanţa euclidiană. Aceasta se defineşte ca fiind rădăcina pătrată a sumei diferenţelor valorilor de pe fiecare dimensiune la pătrat dintre punctul nou (Xnou) si punctul existent (Xi). Aceasta este distanţa clasică dintre două puncte.

Alte măsuri de distanţe populare sunt:

* Distanţa Hamming: Calculează distanţa dintre doi vectori binari;
* Distanta Manhattan: Calculează distanţa dintre doi vectori reali, sumând valoarea absolută a diferenţelor;
* Distanta Minkowski: este o generalizare a distanţei euclidiene şi a distanţei Manhattan.

Pentru fiecare problemă trebuie căutată masura de distanţă care se potriveşte cel mai bine în funcţie de proprietăţile datelor cu care se lucrează. Atunci când nu se cunoaşte cea mai bună măsură de distanţă, se pot experimenta mai multe metrici de distanţă cu diferite valori pentru K şi se va alege modelul care va avea cea mai bună acurateţe.

Spre exemplu, distanţa Euclidiană este o metrică bună atunci când instanţa de intrare are atribute numerice la o scală asemănătoare (ex. lungimea şi lăţimea dreptunghiurilor). Distanţa Manhattan este o metrică bună pentru cazul în care variabila de intrare nu are atribute de acelaşi tip (ex. vârsta, sexul, greutatea unei persoane).

Valoarea cea mai bună pentru K poate fi găsită prin încercarea mai multor valori şi selectarea celei mai bune. Este o buna idee sa se foloseasca valori mai mici decat 21.

Complexitatea algoritmului KNN creşte pe măsură ce cresc şi datele de antrenare. Pentru seturi de date foarte mari, KNN va alege random un eşantion din care va calcula cei mai apropiaţi K vecini.

Când se foloseşte KNN pentru clasificare, predicţia va fi calculată în funcţie de clasa care are cele mai multe apariţii în cei mai apropiaţi K vecini. Fiecare instanţă, în esenţă, votează pentru clasa căreia îi aparţine şi clasa cu cele mai multe voturi câştigă.

Se poate calcula şi probabilitatea cu care o instanţă aparţine unei clase. Spre exemplu, pentru o problemă de clasificare binară: P(barbat) = Numărul de instanţe aproapiate de tipul bărbat raportat la numarul total de vecini, K.

Daca există un număr par de clase, atunci este o bună idee ca valoarea lui K să fie impară, pentru a evita situaţiile în care două clase pot avea acelaşi număr de voturi, situaţie în care apare un caracter random în algoritm. Iar dacă avem un număr impar de clase, este o buna idee ca numărul de vecini pe care-i vom lua in considerare (K) sa fie un numar par.

PRE-PROCESAREA DATELOR PENTRU KNN

* Re-scalarea datelor: KNN funcţionează bine dacă toate datele sunt la aceeaşi scală. Se recomandă ca datele să se normalizeze într-un interval cuprins între 0 şi 1.
* Ştergerea instanţelor care nu au toate atributele: dacă lipsesc atribute, înseamnă că nu se poate calcula similaritatea cu noile instanţe. Acest lucru înseamnă că aceste date oricum nu pot fi folosite şi este natural, să le scoatem din setul de date.
* Numarul de dimensiuni: KNN se pliază pentru problemele în care datele sunt dispunse într-un spaţiu cu dimensiuni cât mai puţine. Se poate aplica şi pe mai multe dimensiuni (de ordinul sutelor de atribute), dar nu va avea aceeaşi acurateţe ca alte tehnici de clasificare. Pentru KNN se poate aplica o reducere de dimensiuni asupra atributelor de intrare.

Mel Frequency Cepstral Coefficient (MFCC)

Primul pas ce trebuie făcut pentru clasificarea melodiilor este acela de a extrage trăsături din ele. Partea cea mai importantă este să încercăm să înţelegem modul în care urechea umană modelează sunetele şi să încercam să reproducem acest proces. Mel Frequency Coefficients( MFCC) sunt caracteristici folosite foarte frecvent în vorbirea automată şi în recunoaşterea vocală.

Aceşti coeficienţi au fost calculaţi prima dată de Davis şi Mermelstein în 1980 şi de atunci sunt un punct de referinţă în domeniul procesării fişiserelor audio.

Atunci când vrem să calculăm aceşti coeficienţi trebuie să trecem prin mai multe etape:

1. Împărţirea în cadre( frame-uri) scurte
2. Pentru fiecare frame se va face o estimare a periodogramei spectrului de energie
3. Se logaritmează fiecare energie
4. Se calculează DCT din valorile logaritmate la pasul anterior
5. Se păstrează primii 13 coeficienţi

**Împărţirea în cadre (frame-uri) scurte**

Un semnal audio se schimbă în continuu, dar pentru simplitate putem presupune că pentru perioade scurte de timp, semnalul audio nu se schimbă foarte mult (această presupunere este făcută din punct de vedere statistic; este evident faptul că se schimbă chiar şi în intervale foarte mici de timp). De aceea vom împărţi semnalul audio în frame-uri de 20-30 milisecunde. Dacă frame-ul ar fi mai mic atunci nu ar fi existat destule date pentru a realiza o estimare rezolabila a spectrului, iar dacă frame-ul ar fi mai mare atunci diferenţaele dintre frame-uri nu ar putea fi comparate .

**Estimarea periodogramei spectrului de energie**

Următorul pas este să calculăm energia pe cele 13 frecvenţe pentru fiecare frame. Acest pas simulează funcţia cohleei, denumită şi melcul membranos. Acesta este un organ din urechea internă umană care vibrează în diferite locuri în funcţie de frecvenţa pe care o are sunetul. În funcţie de locul în care vibrează cohleea, diferiţi nervi se activează şi primesc informaţii pe care le vor transporta către creier. Periodograma face cam acelaşi lucru, identificând frecvenţa din fiecare frame.

Estimarea făcută de periodogramaă încă mai conţine multe informaţii nefolositoare. În special cohleea nu poate face diferenţa dintre două frecvenţe apropiate. Acest efect devinte din ce în ce mai pronunţat pe măsură ce frecvenţa creşte. Din acest motiv, vom grupa mai multe estimări şi le vom aduna pentru a avea o idee despre cât de mult variază energia în diferite regiuni ale melodiei. Acest lucru este făcut de filtrarea Mel. Filtrarea mel cuprinde o grupare de filtre: primul filtru ne va spune cât de multă energie exista aproape de 0 Hertz. Cu cât frecvenţa creşte, filtrele se adaptează astfel încât diferenţa de energie să nu mai fie o problemă.

**Se logaritmează fiecare energie**

Dupa ce s-a facut filtrarea energiei din sunetul audio, trebuie sa le logaritmam. Aceasta logaritmare se face pentru a ne apropia cat mai mult de modul in care functeaza urechea umana: noi nu auzim intensitatea sonora pe o scala liniara. In general, pentru a dubla volumul unui sunet trebuie sa fie consumata de pana la 8 ori mai multa energie. Acest lucru inseamna ca variatiile mari de energie pot sa nu sune la fel de diferit daca sunetul era mult mai intens la inceput. Aceasta operatie face ca atributele noastre sa fie mai asemanatoare cu ceea ce auzim.

**Se calculează DCT**

Ultimul pas este o transformare numita DCT (transformarea discreta a cosinusului) aplicata peste valoriloe obtinute la pasul anterior. Exista cel putin 2 motive mari pentru care se executa aceasta operatie. In primul rand filtrele pe care le-am aplicat se suprapun si sunt destul de corelate intre ele. DCT reuseste sa decoreleze energiile. In al doilea rand mfccs contin doar 13 valori, nu 26. Acest lucru se intampla deoarece 13 coeficienti contin schimbari foarte bruste si acest lucru afecteaza procesarea ulterioara.

Genuri muzicale

Un gen muzical este o categorie conventionala care identifica mai multe piese muzicale ce impartasesc un set de traditii si conventii. Trebuie sa se faca distinctia dintre forma, stilul muzical si genul muzical, chiar daca in practica acesti termeni sunt folositi pentru acelasi inteles. Studiile academice recente au argumentat ca clasificarea melodiilor dupa gen nu poate avea o acuratete perfecta.

Este dificil sa identificam exact in timp cand muzica a inceput sa fie clasificata pe genuri, dar cert este ca secolul trecut si cu siguranta si in zilele noastre aceasta clasificare le domina pe celelalte. Clasificarea melodiilor dupa gen este o problematica dificila. In primul rand, un gen muzical este foarte vast si poate cuprinde multe variatii. Spre exemplu daca o persoana spune ca ii place muzica rock, nu ai de unde sa stii daca se refera la The Beatles, Bob Dylan sau Jimi Hendrix – dar acesti artisti variaza foarte mult ca stil. Sau daca o persoana prefera muzica pop, cum poti sti daca-i place Michael Jackson sau Justin Bieber?

Genurile muzicale sunt create mai degraba de societate decat de caracteristicile muzicii. Atunci cand se inregistreaza un nou album sau o noua melogie, I se atribuie un anumit gen pentru a tinti spre o anume audienta, spre o anumita varsta.

Muzica poate fi impartita pe genuri in foarte multe feluri. Natura artistica a muzicii inseamna ca aceste clasificari sunt, de cele mia multe ori, subiective si controversate, iar unele genuri pot sa se suprapuna partial. Sunt chiar si variatii academice a definitiei termenului gen (muzical).

Probleme cu clasificarea genurilor muzicale

Vrei sa asculti muzica noua dar ai sfarit prin a te uita la colectia ta veche cu meleodii?

Aprlicatiile existente ca si Spotify, Last.fm si Pandora face recomandari pe baza similaritatii dintre artisti.

Musicovery foloseste alta abordare pentru a recomanda melodii: creeaza o retea cu noduri si legaturi de la un artist la altul si face distinctia dintre starile pe care acestia le transmit: Calm, Energic, Pesimist, Optimist.

De-a lungul timpului au avut loc dezbateri cu privire la cea mai buna abordare a clasificarii melodiilor in functie de gen si s-a ajuns la concluzia ca este o chestiune subiectiva. De cele mai multe ori genurile nu corespund nici macar cu sunetul muzicii, ci cu anul si cultura in care au aparut. Unii cercetatori explica performanta scazuta a metodelor de clasificare prin natura genurilor muzicale de a se suprapune [Tzanetakis and Cook, 2002]

Limbajul de programare ales

Din totdeauna a fost o provocare alegerea unui limbaj de programare care sa fie capabil sa le surclaseze pe celelalte, in special cand vine vorba de Data Science. Acest lucru devine si mai dificil atunci cand vrei sa lucrezi la un prim proiect si nu stii ce limbaj de programare sa inveti. Limbajele de programare precum R, Python, Octave, Matlab, Julia etc. ofera o serie de capabilitati care sa faca mai usoare operatiile de analiza a datelor intr-un mod mult mai bun decat limbajele de programare traditionale ca Java, C++, C etc.

Siva Prasad Katru lucreaza ca Chief Architect pentru institutul de tehnologie din India. El a incercat sa clasifice limbajele de programare in functie de diferit metrici, astfel incat sa-l identifice pe cel mai bun.

Pentru a ajunge la o concluzie obiectiva, a create o scala de la 0 la 5 si fiecare limbaj a obtinut o nota, pentru diferite capabilitati precum: viteza de executie, cat de greu este sa inveti acel limbaj, uneltele pentru analiza datelor oferite, metode de vizualizare, unelte de dezvoltare(IDE), cat de usor este sa integrezi o aplicatie noua cu una existenta si oportunitatile locurilor de munca pe piata.

**POZA**

Analizand rezultatul comparatiei putem observa putem spune fara sa ezitam ca “Python conduce detasat, doar R vine din urma destul de puternic”.

DE CE scikit-learn ?

Din multitudinea de utilitare/biblioteci de analiza avansata a datelor: Spark (cu libraria MLLIB), R, scikit-learn, GraphLab etc. Am optat pentru scikit-learn pentru ca e usor de folosit, foarte bine documentat si contine implementari a algoritmilor populari din invatare automata.

Documentatie foarte buna

Principalul motiv pentru care am ales sa folosesc scikit-learn a fost datorita documentatiei foarte bine pusa la punct. Persoanele care doresc sa contribuie la acest proiect open-source sunt nevoite sa scrie si documentatia aferenta, care trebuie sa contina obligatoriu exemple de scripturi care sa ruleze pe seturi de date mici. Pe langa documentatie comunitatea de persoane care s-a format este consecventa si contributiile noi sunt folositoare si de calitate. De asemenea ei sunt incurajati sa dezvolte, in permanenta, noi teste.

Modelele sunt alese si implementate de o echipa de experti

Comunitatea de persoane care contribuie la scikit-learn include si experti din domeniul invatarii automate si a dezvoltarii de software. Si cativa din ei, chiar pun o parte din timpul lor profesionl pentru a dezvolta acest proiect.

Acopera majoritatea ariilor din domeniul invatarii automate

Lista cu uneltele disponibile in scikit-lear cuprinde foarte multe domenii din invatarea automata (precum clasterizare, clasificare, regresi etc). Si din moment de scikit-learn este dezvoltat de o comunitate numeroasa de experti in invatatre automata, este foarte posibil ca tehnicile oferite sa se inmulteasca in urmatoarea perioada.

Utilizatorii nu sunt nevoiti sa aleaga din implementari a aceluiasi algoritm (o problema cu care utiliztorii limbajului R se confrunta). Pentru a-i ajuta pe utilizatori sa gaseasca modelul care se potriveste cel mai bine pe problema lor, Andreas Muller a creat o diagrama pentru acestia.

**POZA**

Focus

Scikit-learn este o biblioteca de invatare automata. Scopul ei este de a oferi un set de algoritmi comuni utilizatorilor de python printr-o interfata consistenta. Acest lucru aduce cu sine alegeri dificile care trebuie facute pentru a sti ce se pliaza pe acest proiect. De exemplu, comunitatea a decis ca algoritmii de Deep- Learning au cateva dependente speciale care ar fi trebuit incluse. Dar s-a decis rezuntarea la ele si implementarea, in schimb, a aunui algortim Multilayer Perceptron.

<http://machinelearningmastery.com/logistic-regression-for-machine-learning/>

<https://en.wikipedia.org/wiki/Music_genre>

<https://www.linkedin.com/pulse/r-vs-python-matlab-octave-julia-who-winner-siva-prasad-katru>

<https://www.oreilly.com/ideas/six-reasons-why-i-recommend-scikit-learn>

AMGC problem

Introducere

Dimensiunea mare de informatii multimedia de pe internet face necesara construirea unor noi tipuri de utilitare automate, capabile sa clasifice melodii in functie de gen, pe baza continutului. Genul unei melodii este unul din cele mai importante caracteristici ce descriu o melodie, si este folosita pentru organizarea colectiilor mari de muzica digitala.

Din perspectiva recunoasterii de modele, lasificarea melodiile in functie de gen este o problema interesanta de cercetare, din moment ce melodiile au o variatie a semnalului audio foarte variata. Un alt aspect interesat este ca clasificarea genurilor este o problema cu mai multe clase. Pentru a putea face clasificari cu mai mult de doua clase putem avea doua abordari:

1. Folosirea tehnicilor care pot clasifica instantele in mai multe clase, in mod natural, producand o separare complexa a spatiului instantelor. Exemple de tehnici pot fi arbori de decizie, K nearest neighbors, retele neuronale etc.

2. Folosirea unei strategii pentru descopunerea problemei de clasificare in mai multe clase intr-o serie de probleme de clasificare binara, folosind clasificatori care vor clasifica spatiul de solutii in mod binar. Exemple de tehnici SVM , clasificare logistica, etc.

In cateva domenii

https://www.researchgate.net/figure/220942923\_fig1\_Fig-1-An-example-illustrating-the-problem-space-decomposition-strategy

Sa ma uit aici:

https://www.researchgate.net/publication/3333877\_Musical\_genre\_classification\_of\_audio\_signals\_IEEE\_Trans\_Speech\_Audio\_Process