**LiDAR duomenimis pagrįstas augmenijos ir elektros linijų klasifikavimas**

**Problemos analizė**

Pagrindinė problema – klasifikuoti objektus iš LiDAR taškų debesies duomenų. Jei yra papildomų duomenų, tokių kaip RGB ar terminiai vaizdai, jie taip pat gali būti naudojami klasifikacijai pagerinti.

Tikslas – aptikti ir klasifikuoti augmeniją (medžius, krūmus, žolę) bei infrastruktūrą, tokią kaip elektros linijos.

**Įvesties duomenys**

* 3D koordinatės (X, Y, Z): Nurodo objektų padėtį ir aukštį.
* Intensyvumo reikšmės: Atspindžio stiprumas, kuris gali padėti atskirti paviršius.
* RGB duomenys (jei yra): Fiksuoja spalvų informaciją, kuri naudinga skirtingų augmenijos tipų atpažinimui.
* Terminiai duomenys (jei yra): Padeda atskirti gyvus objektus (augmeniją) nuo negyvų objektų (infrastruktūros).

**Išvestis:**

Klasifikuoti taškai: Sistema grąžina taškų debesį, kuriame kiekvienam taškui priskirta klasė:

* 0 – Žolė
* 1 – Krūmas
* 2 – Medis
* 3 – Elektros linijos

Išvesties formą sudaro tas pats taškų debesis su papildoma informacija apie kiekvieno taško klasę. Klasifikacijos rezultatai gali būti pateikiami JSON formatu arba kaip CSV failas, kur kiekvienas taškas turi priskirtą klasę.

**Alternatyvių sprendimų analizė**

|  |  |  |
| --- | --- | --- |
| **Sistema** | **Privalumai** | **Trūkumai** |
| **Support Vector Machine (SVM) su RBF kerneliu** | * Efektyviai tvarko nelinijines ribas. * Gerai tinka aukštos dimensijos duomenims, kaip LiDAR. * Atsparus triukšmui, kai tinkamai sureguliuotas. | * Reikalingas kruopštus hiperparametrų parinkimas. * Gali būti lėtas su dideliais duomenų rinkiniais. |
| **Sprendimų medžiai arba atsitiktiniai miškai** | * Lengvai interpretuojamas. * Mažesnė per didelio pritaikymo rizika (atsitiktiniai miškai). | * Mažiau efektyvūs su aukštos dimensijos duomenimis. * Gali per daug prisitaikyti prie duomenų (overfitting). |
| **k-NN (k arčiausių kaimynų) algoritmas** | * Paprastas algoritmas. * Nereikalauja mokymo etapo. | * Neefektyvus dideliuose ar aukštos dimensijos duomenyse. * Jautrus triukšmui ir nereikalingiems požymiams. |
| **CNN (konvoliuciniai neuronų tinklai)** | * Puikiai tvarko vaizdų duomenis. * Automatiškai išskiria svarbius požymius iš vaizdų. | * Reikalauja daug pažymėtų duomenų. * Didelės kompiuterinės išlaidos. * Mažiau efektyvus LiDAR taškams. |

**Kodėl pasirinktas SVM su RBF kerneliu?**

Atsižvelgiant į užduoties specifiką, SVM su RBF kerneliu yra tinkamiausias pasirinkimas dėl šių priežasčių:

1. **Gebėjimas valdyti nelinijines ribas:**
   * Duomenys, ypač augmenijos ir infrastruktūros atskyrimas, yra nelinijiški, todėl RBF branduolys leidžia efektyviai išspręsti šias problemas.
2. **Aukštos dimensijos duomenys:**
   * LiDAR duomenys yra aukštos dimensijos, ir SVM algoritmas yra pritaikytas dirbti su tokiais duomenimis.
3. **Mažesni duomenų rinkiniai:**
   * Jei turime mažesnį pažymėtų duomenų kiekį, SVM algoritmas vis tiek gali užtikrinti gerą klasifikaciją, ko negalime tikėtis iš neuroninių tinklų.
4. **Triukšmo valdymas:**
   * Tinkamai sureguliuotas SVM algoritmas gali būti atsparus triukšmingiems duomenims, kurie dažnai pasitaiko LiDAR taškų debesyse.

**Galimos problemos su duomenimis**

1. **Triukšmo taškai:**
   * Problema: LiDAR duomenys dažnai turi triukšmo taškų – klaidingus ar nereikšmingus duomenis, tokius kaip netikslūs matavimai, ore esantys dalelės, paukščiai ar klaidingi atspindžiai.
   * Sprendimas: Reikėtų naudoti filtravimo metodus triukšmui pašalinti. Pavyzdžiui, galima pritaikyti aukščio filtrus, kurie pašalina taškus, kurie yra per aukštai arba per žemai pagal realų pasaulio objektų kontekstą.
2. **Duomenų netolygumas ir išsisklaidymas:**
   * Problema: LiDAR taškų debesyje taškai gali būti labai netolygiai išdėstyti – kai kuriose vietose jie gali būti labai tankūs, o kitur – reti. Tai apsunkina vienodą objektų atpažinimą.
   * Sprendimas: Gali prireikti taškų retinimo (subsampling) metodų arba interpolacijos, kad užtikrintumėte vienodą taškų pasiskirstymą, taip pat galima atlikti duomenų normalizavimą.
3. **Didelis duomenų kiekis ir kompiuteriniai resursai:**
   * Problema: LiDAR duomenys gali būti labai dideli ir reikalauti daug kompiuterinių resursų tiek apdorojimui, tiek treniravimui.
   * Sprendimas: Duomenų dydį galima sumažinti naudojant taškų debesies retinimą (downsampling) arba ribojant analizuojamą plotą. Be to, reikia apsvarstyti galimybę dirbti su duomenų porciomis arba suskaidyti duomenis į mažesnes dalis.
4. **Balansavimo problema tarp klasių:**
   * Problema: Klasės, pvz., elektros linijos, gali būti labai mažai atstovaujamos lyginant su medžiais ar žole, dėl ko modelis gali per daug pritaikyti (overfitting) labiau dominuojančioms klasėms.
   * Sprendimas: Reikėtų subalansuoti duomenis arba naudoti specialius algoritmus, pvz., SMOTE (Synthetic Minority Over-sampling Technique), kurie padidina mažumų klasių pavyzdžių skaičių.
5. **Požymių reikšmės, nesusijusios su užduotimi:**
   * Problema: Kai kurie požymiai gali būti nesusiję su klasifikavimo užduotimi ir gali padidinti klaidų tikimybę.
   * Sprendimas: Vykdyti požymių atranką (feature selection), siekiant pasirinkti tik svarbiausius požymius klasifikavimui. Galima naudoti tokius metodus kaip koreliacijos analizė ar principal component analysis (PCA).

**Ką reikia padaryti, kad paruošti duomenis sistemai:**

1. **Triukšmo pašalinimas:**
   * Taikyti filtravimo metodus, tokius kaip statistinis filtravimas arba filtras pagal taškų tankumą, kad pašalintumėte triukšmingus ir nereikšmingus taškus.
2. **Požymių išskyrimas ir normalizavimas:**
   * Išgauti reikšmingus požymius (X, Y, Z koordinatės, intensyvumas, RGB ir terminių duomenų reikšmės).
   * Normalizuoti požymius į tą pačią skalę, kad modelis būtų mažiau jautrus dideliems požymių mastelių skirtumams.
3. **Požymių atranka:**
   * Atlikti požymių atranką arba sumažinimą, kad išvengtumėte nereikalingų požymių ir pagerintumėte modelio efektyvumą bei tikslumą.
4. **Duomenų žymėjimas:**
   * Naudoti rankiniu būdu pažymėtą duomenų dalį kaip mokymosi bazę. Jeigu įmanoma, naudoti pusiau prižiūrimus mokymo metodus, kad sumažintumėte rankinio žymėjimo kiekį.

**Veiksmų eiliškumas**

1. **Duomenų paruošimas:**
   * Nuskaityti LiDAR taškų debesį.
   * Filtruoti triukšmą ir pašalinius taškus (pvz., neteisingus matavimus, oro daleles, klaidingus atspindžius).
   * Normalizuoti požymių reikšmes (aukštis, intensyvumas, spalvos, šiluma), kad jos būtų tinkamos SVM mokymui.
2. **Požymių išskyrimas:**
   * Iš kiekvieno taško išgauti X, Y, Z koordinates, intensyvumo reikšmes, ir (jei yra) RGB bei šiluminius duomenis.
   * Sukurti požymių vektorius, naudojamus kaip įėjimus SVM modeliui.
3. **Duomenų padalijimas:**
   * Padalinti duomenų rinkinį į mokymo ir testavimo rinkinius. Paprastai 70% duomenų skiriami mokymui, o likę 30% – testavimui.
4. **SVM modelio mokymas:**
   * Pritaikyti RBF kernelio SVM modelį. Naudojant mokymo duomenis, SVM ieško optimalaus hiperlėktinio plokštumos, kuri geriausiai atskirtų duomenų klases.
   * Nustatyti hiperparametrus:
     + C (reguliavimo parametras) balansuoja tarp marginos didinimo ir klaidingų klasifikacijų mažinimo.
     + γ valdo atstumą, kurį kernelis veikia, ir taškų įtaką klasifikacijai.
5. **Modelio testavimas:**
   * Naudoti testavimo rinkinį, kad patikrintumėte modelio našumą ir tikslumą.
   * Apskaičiuoti klasifikavimo metrikas: tikslumas, atšaukimo ir vidurkis tarp tikslumo ir atšaukimo.
6. **Klasifikacija:**
   * Naudoti išmokytą SVM modelį visam taškų debesies rinkiniui klasifikuoti.
   * Kiekvienam taškui priskiriama viena iš klasių (0 – žolė, 1 – krūmas, 2 – medis, 3 – elektros linijos).
7. **Rezultatų išsaugojimas ir vizualizacija:**
   * Išsaugoti klasifikacijos rezultatus į JSON arba CSV failą, kuriame kiekvienas taškas turi priskirtą klasę.
   * Vizualizuoti rezultatus, naudojant skirtingas spalvas kiekvienai klasei:
     + Žolė – žalia,
     + Krūmai – geltona,
     + Medžiai – tamsiai žalia,
     + Elektros linijos – pilka.

**Formulės ir jų paaiškinimas**

1. **RBF (Radialinės Bazės Funkcija) branduolys:**

RBF branduolys naudojamas įėjimo duomenims perkelti į aukštesnės dimensijos erdvę. RBF branduolio K(xi,xj) formulė

* + xi, xj​ yra požymių vektoriai (pvz., LiDAR duomenys, vaizduojantys mišką, žolę ar elektros linijas).
  + ||xi - xj|| yra Euklido atstumas tarp dviejų duomenų taškų.
  + yra parametras, kuris valdo RBF branduolio plotį (jį galima reguliuoti treniruojant modelį).

Naudojimo atvejis: Ši formulė naudojama apskaičiuoti duomenų taškų panašumą. Kuo didesnis panašumas, tuo arčiau taškai yra požymių erdvėje.

1. **SVM sprendimo funkcija**

SVM sprendimo funkcija naudojama duomenų atskyrimui į skirtingas klases. Sprendimo funkcija f(x) yra tokia:

* + xi​ yra atraminiai vektoriai (duomenų taškai, kurie apibrėžia sprendimo ribą).
  + αi ​ yra Lagranžo daugikliai (jie nustatomi SVM optimizavimo proceso metu).
  + yi ​ yra klasės žymė (klasifikacijoje yi gali būti +1 arba -1, pvz., miškas arba ne miškas).
  + K(xi,x) yra branduolio funkcija.
  + b yra poslinkio parametras (taip pat nustatomas treniruojant modelį).

Naudojimo atvejis: Ši formulė padeda klasifikuoti naujus duomenų taškus, vertinant, kurioje sprendimo ribos pusėje jie yra (pvz., klasifikuojant, ar naujas LiDAR duomenų taškas priklauso "miškui" ar "ne miškui").

1. **SVM su minkšta riba (Soft Margin SVM):**
   * yra svorių vektorius, apibrėžiantis sprendimo ribą.
   * C yra reguliavimo parametras (kontroliuoja kompromisą tarp ribos maksimizavimo ir klaidų leidimo).
   * yra klaidų kintamieji (jie rodo, kiek duomenų taškas pažeidžia ribą).

Naudojimo atvejis: Realiuose LiDAR duomenyse dažnai pasitaiko triukšmas ir klaidos (pvz., klaidingai klasifikuoti medžiai kaip elektros linijos). Minkštos ribos SVM padeda atsižvelgti į šias klaidas ir tuo pačiu optimizuoti sprendimo ribą.

**Tikėtinas rezultatas**

Taškų klasifikacija pateikiama tam tikru formatu, pvz., JSON arba CSV, kuriame kiekvienas taškas turės papildomą lauką, nurodantį klasę, kuriai jis priskiriamas. Šie duomenys gali būti naudojami tolimesnei analizei arba vizualizavimui.

Vizualiai, taškų debesies rezultatai galėtų būti rodomi kaip spalvotų taškų rinkinys, kur kiekvienai klasei yra priskiriama unikali spalva, pvz.:

* Žalia spalva – žolė,
* Geltona spalva – krūmai,
* Tamsiai žalia spalva – medžiai,
* Pilka spalva – elektros linijos.

Ši sistema suteiks galimybę lengvai atskirti įvairias augmenijos rūšis nuo infrastruktūros objektų, tokių kaip elektros linijos, todėl ji gali būti naudojama stebint ir valdant rizikas, pvz., elektros linijų priežiūrą ir jų apsaugą nuo per arti esančios augmenijos.