Tema Învățare Automată

Partea 1

Dată publicare: 30/10/2024	Dată limită rezolvare: 27/11/2024

1. Descriere generală

În practica de zi cu zi a unui inginer sau cercetător în domeniul învățării automate intră frecvent următoarele trei aspecte:

- Vizualizarea și “explorarea” datelor unei probleme (Exploratory Data Analysis)
- Încercarea de a extrage atribute ale datelor problemei pentru a fi utilizate în obiectivul de analiză ales (e.g. clasificare, regresie, detectie de anomalii)
- Evaluarea mai multor modele pentru găsirea soluției celei mai bune pentru problema dată

Sarcinile voastre de lucru vor solicita utilizarea de biblioteci de **vizualizare a datelor (crearea de diagrame)**, **extragerea de atribute (feature extraction)** pentru folosirea algoritmilor de clasificare discutați la curs, precum și **utilizarea unor modele** de machine learning.

2. Descrierea Seturilor de Date

Aveți la dispoziție două seturi de date cu imagini:

- Fashion-MNIST - un set de date cu 70000 de imagini grayscale de tip thumbnail (32 x 32) reprezentând 10 tipuri de elemente de vestimentație (e.g. bluze, cizme, pantaloni)
- Fruits-360 - un set de date cu ~55000 de imagini RGB cu 80 de tipuri diferite de fructe singulare

Seturile de date au deja împărțire în date de antrenare (train) și de testare (test).
Obiectivul în fiecare set de date este cel de clasificare corectă a imaginii în categoria corectă.

3. Extragere de Atribute din Imagini

Fiind vorba de imagini, extragerea de atribute se face în termenii spațiului de culori, textură și geometrie a formelor.
În cele ce urmează descriem pe scurt și furnizăm referințe de exemplificare pentru o serie de metode de **extragere de atribute** din imagini, cunoscute în literatura de specialitate.

PCA (Principal Component Analysis). Analiza Componentelor Principale este o tehnică de reducere a dimensionalității care transformă datele de o dimensionalitate ridicată într-o formă de dimensiune mai mică, păstrând în același timp cea mai mare variație. PCA identifică direcțiile în care datele variază cel mai mult (componentele principale) și proiectează datele pe aceste axe. Pe cazul imaginilor, cele mai relevante componente principale identificate printr-o procedură ca PCA pot fi "responsabile" de atribute din imagine precum: culoare sau textură dominantă, forme sau muchii de o anumită orientare preponderente în imagine. Tutorial explicativ disponibil aici.

HOG (Histogram of Oriented Gradients) este o tehnică de extragere a unor descriptori ce se concentrează pe forma unui obiect, numărând aparițiile orientărilor gradientului în regiuni locale. HOG desparte imaginea în regiuni mici (celule) și acumulează informațiile despre gradient în histograme care reflectă distribuția direcțiilor contururilor.
Tutorial: https://builtin.com/articles/histogram-of-oriented-gradients

ORB (Oriented FAST and Rotated BRIEF) este un algoritm proiectat pentru a identifica și descrie eficient caracteristici cheie din imagini (regiuni distincte și ușor identificabile). ORB combină metoda FAST pentru detectarea punctelor cheie cu BRIEF pentru descrierea acestora, fiind invariant atât la rotație, cât și la variațiile de iluminare. ORB oferă o alternativă mai rapidă și eficientă în comparație cu SIFT, cu performanțe comparabile în detecția caracteristicilor. Tutorial explicativ disponibil aici.

Metode pe baza de contur furnizează atribute ce descriu forma unui obiect, precum perimetrul conturului, suprafața descrisă de contur, soliditate, circularitate, Hu Moments și altele. O serie de atribute pe baza de formă se pot obține folosind biblioteca OpenCV sau Scikit Image, îndeosebi metodele din categoria regionprops / regionprops_table.

Metodele de extragere de atribute pot fi și combinate între ele. Un exemplu este prezentat în acest tutorial. Acesta utilizează următoarea combinație a metodelor prezentate mai sus:

- Se folosește metoda Scale Invariant Feature Transform (SIFT - similară cu ORB) pentru a găsi puncte cheie în imagine
- În jurul punctelor cheie se definește o regiune rectangulară, având punctul cheie drept mijloc
- Se calculează atribute ale regiunilor identificate folosind metoda **HoG**
- Se aplică algoritmul **k-means** pentru a grupa toate atributele obținute de la pasul anterior și a obține un *vocabular* de atribute reprezentative

- Se construiește o histogramă a distribuției atributelor reprezentative pe fiecare imagine din setul de date; histograma obținută devine **seria de atribute a acelei imagini**
- Se folosește histograma ca intrare pentru algoritmi de clasificare precum SVM sau GradientBoosting

4. Cerințe

4.1. Extragerea de atribute (Feature Extraction) [3p]

Definiți fluxul vostru de extragere a atributelor, **bazat pe cel puțin două metode** din setul celor exemplificate mai sus. Documentați în raportul temei pașii pe care îi efectuați, împreună cu o justificare pentru metodele de extragere de atribute alese. Justificarea voastră trebuie să facă referire și la rezultatele obținute.

4.2. Vizualizarea atributelor extrase [2p]

Pentru a înțelege mai bine problema și efectul atributelor de imagine extrase în cazul fiecărui dataset, o primă etapă este cea de aplicare / construire a unor metode de ***vizualizare a datelor*** și/sau de ***raportare a distribuțiilor de valori*** pe fiecare atribut folosit în predicție.

Analize minime obligatorii

1. **Analiza echilibrului de clase**

Realizați un grafic al frecvenței de apariție a fiecărei etichete (clase) în seturile de date de antrenare / test, folosind **bar plot** / **count plot**.
Pentru realizarea unor astfel de bar plots puteți folosi mai multe biblioteci:

- Folosind biblioteca seaborn pentru barplot sau countplot
- Direct dintr-un DataFrame Pandas folosind pandas.DataFrame.plot.bar

2. **Vizualizarea cantitativă / calitativă a efectului de extragere a atributelor**

În funcție de metodele de extragere a atributelor alese, realizați:

- **vizualizări cantitative** (dacă se aplică) sub formă de tabel sau grafic la nivel de statistică per clasă sau per întreg setul de date
- **vizualizări calitative** ale aplicării metodei de extragere pe **câte o imagine din fiecare clasă (pentru setul de date Fashion-MNIST)** si pe **câte o imagine din cele mai numeroase 10 clase (pentru setul de date Fruits-360).** Exemple de vizualizări calitative includ:

Exemple de vizualizări cantitative:

- Gradul de varianță cumulativă explicată de numărul ales de componente principale
- O analiză statistică a mediei **histogramelor de atribute per clasă** pentru metoda de extragere de atribute ce combină descriptori SIFT, k-means si histograme de atribute

Exemple de vizualizări calitative:

- Pentru metoda PCA: Vizualizarea imaginilor reconstruite folosind componentele principale selectate
- Vizualizarea punctelor cheie identificate în imagini prin suprapunerea lor peste imaginea inițială (exemple de cod în tutorialul despre ORB sau documentația despre ORB din OpenCV)
- Trasarea formelor de contur identificate pentru obiecte sau a metodelor de aproximare a conturului (e.g. Convex Hull, Ellipse approximation) peste imaginile inițiale (vedeți un exemplu de afișare a conturului aici)

Atenție!

- Se cere cel puțin o interpretare calitativă per metodă de extragere a atributelor aleasă (a se vedea cerința 4.1).
- Interpretările calitative se fac pentru **fiecare set de date în parte** (FashionMNIST, Fruits-360)
- Vizualizările cantitative și calitative **trebuie însoțite de un comentariu asupra** utilității anticipate sau al înțelesului desprins asupra problemei din utilizarea respectivelor atribute pentru imagini din diferite clase.

4.3. Standardizarea și selecția atributelor [1p]

Folosind mai multe metode de extragere a atributelor din cele exemplificate în Secțiunea 3, numărul atributelor este posibil să ajungă ridicat. Mai mult, e posibil ca atributele extrase să aibă valori numerice semnificativ diferite între ele (e.g. rapoarte subunitare la un atribut precum soliditatea vs. valori de ordinul sutelor ca suprafață în număr de pixeli).

Folosiți proceduri de standardizare a datelor ca etapă de preprocesare a datelor înainte de antrenarea unui clasificator, în vederea uniformizării valorilor numerice aferente fiecărui tip de atribut.

Frecvent se întâmplă ca nu toate atributele să aibă o contribuție importantă în cadrul predicției. Ca atare, **investigați aplicarea tehnicilor de selectare a atributelor (eng. Feature selection)** oferite în scikit-learn. Folosiți cel puțin una din metodele ***Variance Threshold*** sau ***Select Percentile***. **Explicați** (măcar intuitiv) diferențele dintre cele 2 seturi (cel inițial și cel redus prin selecție).

4.4. Utilizarea algoritmilor de Învățare Automată [4p]

Pentru efectuarea taskului de clasificare peste fiecare set de date veți folosi următorii algoritmi:

- LogisticRegression - folosiți implementarea din scikit-learn
- SVM - folosiți implementarea din scikit-learn
- RandomForest - folosiți implementarea din scikit-learn
- GradientBoosted Trees - folosiți implementarea din biblioteca xgboost

Fiecare algoritm din cei propuși are o serie de **hiper-parametri** care influențează funcționarea acestuia. Pentru a găsi valorile potrivite pentru aceștia veți folosi o procedură de **căutare a hiper-parametrilor.**
Setul minim de hiper-parametri de căutat este:

- LogisticRegression - parametru *C* de regularizare, metodologia de clasificare multinomială (parametrul multi_class ales între “ovr” sau “multinomial”)
- SVM: tipul de kernel, parametru *C* de regularizare
- RandomForest: numărul de arbori, adâncimea maximă a unui arbore, procentul din input folosit la antrenarea fiecărui arbore
- GradientBoostedTrees: numărul de arbori, adâncimea maximă a unui arbore, learning rate

Căutarea hiper-parametrilor se poate face în prin două metode:

- Folosind un **set de validare**
- Folosind procedura de **Randomized Search with Cross-Validation**

Atenție! Procedura de căutare prin Cross Validation **poate dura foarte mult**, dacă setul de date este mare, alegeți multe fold-uri și căutați după mulți parametri. Analizați dacă puteți restructura seturile voastre de date, astfel încât să obțineți un **set de validare** pe care să-l folosiți pentru procedura de potrivire a hiper-parametrilor, renunțând astfel la Cross-Validation.

Evaluarea algoritmilor
În raportul vostru trebuie să prezentați următoarele:

- Rezultatul procedurii de feature selection: numărul total de feature-uri considerate și numărul total de feature-uri utilizate la antrenare (ca urmare a procedurii de feature selection). În cazul în care acestea diferă, explicați (intuitiv) de ce, cu referire concretă la coloanele în cauză.
- Pentru fiecare algoritm, realizați un tabel în care să prezentați **media și varianța** pentru ***acuratețea generală de clasificare***, ***precizie / recall / F1*** **(la nivel de maxim 3 zecimale) la nivelul fiecărei clase în parte**
    - Pe linii va fi indexată configurația de hiper-parametri rezultată din procedura de căutare
    - Pe coloane vor fi prezentate metricile cerute
    - **Relevați prin bolduire** valorile maxime pentru fiecare metrică
- Pentru **cea mai bună variantă a hiper-parametrilor,** pentru **fiecare algoritm**, și **pentru fiecare dataset în parte** realizați o **matrice de confuzie** (**ca diagramă color-coded -** e.g. folosind ConfusionMatrixDisplay) peste clase.

5. Predarea temei

Tema va fi încărcată pe Moodle însoțită de un raport sub formă de fișier PDF, care include:

- **Cerința 4.2** - cuprinde toate vizualizările și statisticile cerute. **Este obligatorie** prezența în text a **unei interpretări / analize** a diagramelor rezultate.
- **Cerința 4.3-4.4** - include raportarea extragerii de atribute și a evaluării algoritmilor de clasificare pentru cele două seturi de date propuse. **Este obligatorie** prezența în text a **unei interpretări / analize** a rezultatelor obținute (e.g. care atribute sunt cele mai predictive, cât de puternic este impactul hiper-parametrilor asupra performanței fiecărui algoritm considerat, care sunt clasele cu cele mai bune predicții).

Rezultatele temei vor fi prezentate în cadrul laboratoarelor de Învățare Automată, **exclusiv pe baza rapoartelor încărcate.**