Nebalansiranost podataka i pomak koncepta

Dubinska analiza podataka

5. predavanje

Pripremio: izv. prof. dr. sc. Alan Jović

Ak. god. 2023./2024.

Sadržaj

- Problem nebalansiranosti podataka
- Pomak koncepta u podacima
  - Analiza tokova podataka

Problem nebalansiranosti podataka

Nebalansiranost podataka

- Engl. *imbalanced learning, class imbalance problem*
- Problem skupa podataka u kojem postoji **neravnoteža (nebalansiranost) u broju primjeraka pojedinih klasa ciljne značajke**
- Neravnoteža u broju primjera pojedinih klasa otežava izgradnju modela koji će jednako dobro klasificirati i **većinske** i **manjinske** klase
- Neravnoteža podataka **jednako pogađa modele strojnog i dubokog učenja**
- U općem slučaju, problem klasifikacije je **višeklasan** (engl. *multiclass*)
- U praksi često kao **predviđanje rijetkih događaja** (engl. *rare event prediction*)

Primjena

- Nebalansirani podaci i rijetki događaji ima vrlo široku primjenu
  - Biomedicina (npr. detekcija rijetkih bolesti), kemija (npr. detekcija patvorenja), financijski menadžment (npr. detekcija prijevare), industrija (npr. kvarovi opreme)...
- Problem: 99%+ vremena stvari su „normalne”, kako predvidjeti rijetki događaj?
  - Potrebna su prošla opažanja, kojih može biti jako malo
  - Predikcija (u smislu budućeg vremena) je moguća ako postoji neka vrsta specifičnog prethodnog pokazatelja (signala, tranzicijskog prozora) koji nagovješta budući rijetki događaj, inače možemo govoriti samo o detekciji

Stupanj nebalansiranosti klasa

- **Ne postoji strogo definirani kriterij** za postojanje nebalansiranosti klasa
- Najčešće, većinske klase treba biti najmanje 100% više od manjinske (omjer 2:1) da postoji nebalansiranost koja može utjecati na sposobnost ispravne klasifikacije
- Stupnjevi nebalansiranosti – **omjer nebalansiranosti** (engl. *imbalance ratio*) primjeraka većinska klasa : manjinska klasa:
  - Do omjera 4 (4:1) – blaga neuravnoteženost (klasifikatori većinom mogu dobro učiti)
  - Do omjera 9 (9:1) – srednja neuravnoteženost (klasifikatori uglavnom imaju poteškoća pri učenju)
  - Do omjera 99 (99:1) – velika neuravnoteženost (klasifikatori najčešće ne uče dobro)
  - Više od 100 (100:1) – rijetki događaji (ili rijetke vrijednosti ciljne klase) (klasifikatori gotovo sigurno ne uče dobro) – u medicini, bolest se smatra rijetkom ako pogađa manje od 0,1% populacije

Stupanj nebalansiranosti klasa

- Klasifikatori koji se susretnu s velikom nebalansiranosti **odabiru uglavnom (ili u potpunosti) većinsku klasu**, preciznost detekcije manjinske klase teži nuli
- Manjinski primjerci se mogu tretirati kao šum većinske klase (klasifikator ne zna što je šum, a što manjinski primjerak)
- **Ako su klase potpuno odvojive**, nebalansiranost ne predstavlja problem, međutim malo gdje su klase potpuno odvojive
- **Visoka dimenzionalnost ili premalo primjeraka, uz nebalansiranost, dodatno otežavaju klasifikaciju**

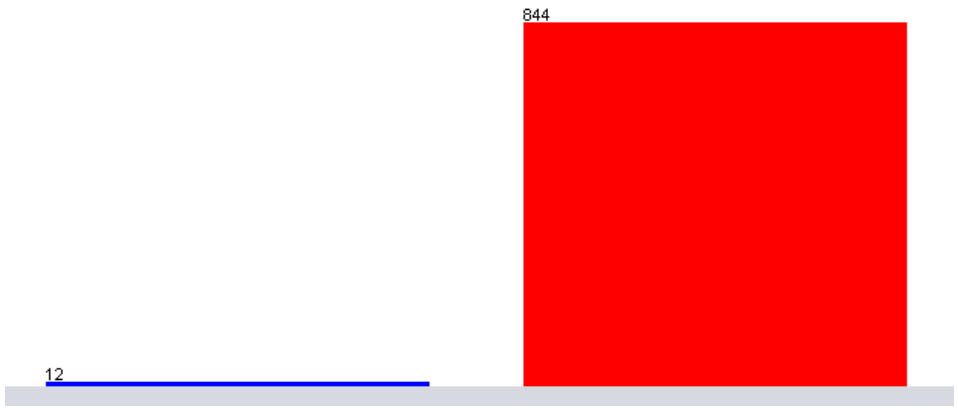
Stupanj nebalansiranosti klasa – primjer

Selected attribute

Name: Outcome Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	Active	12	12.0
2	Inactive	844	844.0

Class: Outcome (Nom) ▼ Visualize All




```
Correctly Classified Instances         843               98.4813 %
Incorrectly Classified Instances        13                1.5187 %
=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall   F-Measure
                 0.000     0.001     0.000       0.000    0.000
                 0.999     1.000     0.986       0.999    0.992
Weighted Avg.    0.985     0.986     0.972       0.985    0.978

=== Confusion Matrix ===

   a   b   <-- classified as
   0  12 |   a = Active
   1 843 |   b = Inactive
```

Model slučajne šume, 100 stabala, 10-fold cross-validation

Izvor: Weka, skup podataka „unbalanced.arff”

Klasifikacija pristupa rješavanju problema nebalansiranosti klasa

- **Pristupi predobrade podataka**
  - **Ponovno uzorkovanje** (engl. *resampling*) – najčešći pristup (u preko 30% članaka)
  - **Odabir značajki i smanjenje dimenzionalnosti**
- **Učenje osjetljivo na cijenu** (engl. *cost-sensitive learning*)
- **Izmjena algoritma strojnog učenja**
- **Primjena ansambala klasifikatora**
- **Korištenje prikladnih mjera vrednovanja modela**

- **Napomena**: pristupi se često **kombiniraju**
- **Programska potpora za Python: https://imbalanced-learn.org/stable/**

Ponovno uzorkovanje

- **Osnovna ideja:** izmijeniti razdiobu primjeraka ciljne značajke po klasama kako bi bila približno jednaka za sve klase (ne zahtijeva se potpuna jednakost)
- Tri najčešća načina provedbe:
  - **Naduzorkovanje** (engl. *oversampling*)
  - **Poduzorkovanje** (engl. *undersampling*)
  - **Hibridni način uzorkovanja** – kombinacija naduzorkovanja i poduzorkovanja

Naduzorkovanje

- Generiranje **novih primjeraka za učenje za manjinsku klasu** (ili klase) kako bi se postigla ravnoteža s većinskom klasom
- Češće se koristi nego poduzorkovanje
- Problematično iz perspektive povećanja broja primjeraka za učenje
  - Nije naročiti problem ako je neuravnoteženost mala

Naduzorkovanje

- Novi primjerci mogu biti generirani:
  - Kopiranjem postojećih primjeraka *n* puta (najlošiji izbor)
  - **Slučajnim naduzorkovanjem s ponavljanjem** postojećih primjeraka (engl. *random oversampling with replacement*)
  - Kao izmijenjeni**, sintetski primjerci** koji uzimaju u obzir postojeće primjerke
- Pristupi generiranju sintetskih primjeraka:
  - Zasnovani na varijantama algoritma **SMOTE** (engl. *Synthetic Minority Over-sampling TEchnique*)
  - Ostali pristupi, npr. zasnovani na nenadziranom učenju, polunadziranom učenju, **ADASYN**, algoritmi rojeva

Slučajno naduzorkovanje s ponavljanjem

- Iz skupa od N_1 primjeraka manjinske klase odabire se u svakoj iteraciji jedan uzorak koji se dodaje u skup primjeraka
- Svaki primjerak manjinske klase ima jednaku vjerojatnost odabira: $p\big(S(x_i)\big) = \frac{1}{N_1}, \forall i \in 1..N_1$, S je odabir
- Primjerak koji je već izabran ima mogućnost ponovno biti izabran (uzorkovanje se radi s ponavljanjem)
- U praksi dosta uspješan pristup, usporediv s puno složenijima

Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explor. Newsl., 6(1):20–29, June 2004.

Naduzorkovanje zasnovano na SMOTE-u

- **SMOTE** – generiranje sintetskih primjeraka na temelju postojećih primjeraka manjinske klase
- Uzima se svaki primjerak manjinske klase i razmatra se njegovih ***k* najbližih susjeda manjinske klase** (prema euklidskoj udaljenosti)
- **Sintetski primjerak generira se na linijskim segmentima u *m*-dimenzijskom prostoru koji povezuju primjerak i njegovih *k* susjeda**
  - Npr. Ako je potrebno napraviti 200% naduzorkovanje, uzimaju se 2 najbliža susjeda ($k = 2$)
  - Za svakog susjeda izračuna se **razlika (udaljenost)** između susjeda i primjerka, pomnoži se **slučajnim brojem između 0 i 1** i **doda primjerku**, čime se dobiva novi sintetski primjerak na liniji koja povezuje primjerak i najbližeg susjeda

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002

Naduzorkovanje zasnovano na varijantama SMOTE-a

- U praksi, najbolji rezultati postižu se za hiperparametar k = 5 (500% naduzorkovanja)
- Algoritam omogućuje definiranje količine naduzorkovanja za svaku klasu, a ako se ono ne odredi, algoritam će generirati primjerke svih manjinskih klasa tako da postigne broj primjeraka jednak jednoj većinskoj klasi
- **Problem SMOTE-a:** izolirani primjerci manjinske klase u prostoru većinske klase stvorit će linijski most preko primjeraka većinske klase
- Neke poznatije varijante SMOTE-a: Borderline-SMOTE i Safe-level-SMOTE

Vidjeti:

Han H., Wang WY., Mao BH. (2005) Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang DS., Zhang XP., Huang GB. (eds) Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11538059_91

Bunkhumpornpat C., Sinapiromsaran K., Lursinsap C. (2009) Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In: Theeramunkong T., Kijsirikul B., Cercone N., Ho TB. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2009. Lecture Notes in Computer Science, vol 5476. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01307-2_43

Alberto Fernández, Salvador García, Francisco Herrera, and Nitesh V. Chawla. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J. Artif. Int. Res. 61, 1 (January 2018), 863–905

Naduzorkovanje zasnovano na ADASYN-u

- **ADASYN** – više sintetskih primjeraka generira se za one primjerke manjinske klase u skupu za učenje koji se **teže uče** (omeđeni su s više primjeraka većinske klase)
  - Za svaki manjinski primjerak računa se omjer r_i između broja većinskih primjeraka i broja susjeda *k* iz *k*-najbližeg susjedstva
  - Taj omjer se normalizira na raspon [0, 1] za sve manjinske primjerke: $\widehat{r_i} = r_i / \sum_{i=1}^{m} r_i$, *m* je broj manjinskih primjeraka
  - Izračuna se broj potrebnih sintetskih primjeraka g_i za svaki manjinski primjerak kao umnožak normaliziranog omjera i broja potrebnih primjeraka koje treba generirati
  - Slučajno se izabere manjinski susjed g_i puta, s ponavljanjem, te se generira novi sintetski primjerak na isti način kao kod SMOTE-a
- **Usporedba SMOTE-a i ADASYN-a:** nijedna metoda nije konzistentno bolja na većem broju skupova podataka

Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.

J. Brandt, E. Lanzen, A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification, Department of Statistics, Uppsala University, 2020.

Poduzorkovanje

- **Uklanjanje primjerka većinske klase (ili klasa) kako bi se postigla ravnoteža s manjinskom klasom**
- Problematično jer se **gubi dio informacije** sadržan u primjercima većinskih klasa
  - Količina gubitka informacije proporcionalna je stupnju neuravnoteženosti klasa
- Nekoliko češćih pristupa:
  - **RUS**, **NCL, TL** (Tomek Links)
- **Metoda slučajnog poduzorkovanja** (engl. *random undersampling*, **RUS**) – većinski primjerci uklanjaju se **slučajnim odabirom bez ponavljanja** sve dok se broj primjeraka većinske klase (ili klasa) ne izjednači (približno) s jednom manjinskom klasom

Poduzorkovanje – metoda NCL

- **Pravilo čišćenja susjedstva** (engl. *Neighborhood Cleaning Rule*, NCL)
  - Koristi pravilo uređenih najbližih susjeda (engl. *Edited Nearest Neighbors*, **ENN**) (Wilson, 1972.) kako bi uklonilo primjerke **većinske** klase
  - Pravilo ENN uklanja primjerak većinske klase ako u njegovih *k*-najbližih susjeda postoji barem jedan primjerak manjinske klase (*defaultni k* = 3, može i *k* = 5)
  - Primjena je najčešće blizu decizijske granice
  - Smanjuje šum u podacima
  - Često dovodi do gubitka informacije

Poduzorkovanje – metoda TL

- **Metoda Tomekovih poveznica** (engl. *Tomek Links*, TL)
  - Tomekova modifikacija metode zgusnutog najbližeg susjeda (engl. *condensed nearest neighbour*, CNN)
  - Dva primjerka **različitih klasa** formiraju Tomekove poveznice ako je euklidska udaljenost između njih manja nego udaljenost između svakog od njih i bilo kojeg drugog primjerka u skupu podataka
  - Varijanta za poduzorkovanje: iz skupa podataka **uklanjaju se primjerci većinske klase koji formiraju Tomekove poveznice s manjinskom klasom**
    - Alternativa: uklanjaju se svi primjerci (i većinski i manjinski) koji formiraju Tomekove veze (ova varijanta se ne koristi za poduzorkovanje, nego samo za čišćenje podataka)

Hibridni postupci uzorkovanja

- **Kombiniranje pristupa poduzorkovanja i naduzorkovanja** (obično, poduzorkovanje ide prvo, a potom naduzorkovanje)
- Jednostavni pristup: **slučajna ravnoteža** (engl. *Random Balance*)
  - Kombinacija slučajnog poduzorkovanja i SMOTE-a
  - Za slučaj dvije klase, odabere se slučajni broj *S* unutar raspona [2, *N*-2], gdje je *N* veličina skupa podataka
  - Većinska klasa se slučajno poduzorkuje dok ne postigne broj primjeraka jednak *S*
  - Manjinska klasa se SMOTE-a dok se ne postigne broj primjeraka *N* - *S*
  - Ako je *S* nekim slučajem veći od broja primjeraka većinske klase, tada se postupak obrne – većinska klasa se SMOTE-a, a manjinska slučajno poduzorkuje
- Složeniji hibridni postupci s dobrim rezultatima: SMOTE+ENN, SMOTE+TL...

J.F. Díez-Pastor, J.J. Rodríguez, C.I. García-Osorio, L.I. Kuncheva, Random balance: Ensembles of variable priors classifiers for imbalanced data, Knowledge-Based Systems 85(2015) 96–111, doi:10.1016/j.knosys.2015.04.022.

Pravilnosti kod ponovnog uzorkovanja

- U općenitom slučaju, pokazuje se da metode naduzorkovanja rade bolje od metoda poduzorkovanja
- Ako se broj manjinskih primjeraka u skupu podataka mjeri u stotinama, tada se pokazuje da je poduzorkovanje većinske klase bolji izbor od naduzorkovanja manjinske klase jer postoji dovoljno informacije o manjinskoj klasi, a računski je manje zahtjevno učiti na poduzorkovanom skupu
- Ako se broj manjinskih primjeraka manji od stotinjak, naduzorkovanje metodom SMOTE se pokazuje najkorisnijim
- Ako je skup za učenje velik (tisuće primjeraka), preporuča se hibridna kombinacija poduzrokovanja i naduzorkovanja
- U ovisnosti od omjera nebalansiranosti, neke metode rade bolje od drugih, ali **nema općenitih pravilnosti**

Odabir značajki i smanjenje dimenzionalnosti

- **Za skupove podataka koji su visokodimenzionalni**, bolji rezultati u slučaju klasifikacije nebalansiranog skupa postižu se prethodnim korištenjem algoritama za odabir značajki i za smanjenje dimenzionalnosti (vidjeti 3. i 4. predavanje)
- Pokazuje se da eliminacijom nebitnih značajki klasifikacija manjinskih primjeraka postaje uspješnija
- Primjena odabira značajki i smanjenja dimenzionalnosti za nebalansirane skupove podataka je vrlo raširena
- **Poanta: najprije provesti odabir značajki / smanjenje dimenzionalnosti, potom pristupiti problemu neuravnoteženosti klasa**

Učenje osjetljivo na cijenu

- Osnovna pretpostavka: **neispravna klasifikacija primjerka manjinske klase u odnosu na većinsku** smatra se da je problematičnija (ima veću cijenu)
- Cijena se izražava u obliku **matrice cijene** (engl. *cost matrix*) u kojoj je određena cijena klasifikacije svake klase u svaku drugu klasu (oblik je kao i konfuzijska matrica)

		Prediction	
		Class i	Class j
True	Class i	0	λ_{ij}
	Class j	λ_{ji}	0

Učenje osjetljivo na cijenu

- Matricu cijene može zadati ekspert na temelju iskustva ili se može odrediti automatski iz podataka
  - Npr. cijena iznosi **1** za većinsku klasu, a iznosi **omjer nebalansiranosti** za manjinsku klasu
    - za skup s 10 primjeraka, od kojih 1 je manjinska, a 9 većinska klasa taj omjer iznosi 9, što znači da penal za neispravnu klasifikaciju manjinske klase iznosi 9 puta toliko koliko za većinsku klasu
- **Izmjena primjeraka za učenje**
  - Svakom primjerku manjinske klase dodjeljuje se **težina** iz matrice cijene kojom ulazi u klasifikaciju
  - Preduvjet za korištenje: algoritam za učenje mora znati raditi s utežanim primjercima

Pristupi izmjene algoritma strojnog učenja

- Ideja: povećanje diskriminatorne snage klasifikatora kako bi bolje razdvajao primjerke različitih klasa s naglaskom na većoj cijeni neispravne klasifikacije manjinskih primjeraka (većinskim primjercima se više tolerira klasifikacija u manjinske primjerke nego obratno)
- Veći broj raznih postupaka (utežani SVM, ANN s izmjenom aktivacijske funkcije, neizrazita pravila, utežani KNN...)
- Primjer: Pretvorba ciljne funkcije (engl. *objective function*) u optimizacijski problem u kojem se više **penalizira neispravna klasifikacija manjinske klase kod stroja s potpornim vektorima (SVM)**
  - Hiperravnina koja razdvaja klase kod SVM modela može biti zakrivljena prema manjinskoj klasi kod nebalansiranog skupa, što degradira klasifikaciju
  - **Utežani SVM** (engl. *weighted SVM*, *cost-sensitive SVM*) – primjerci imaju težine proporcionalne distribuciji klasa, a hiperparametar *C* (regularizacijski hiperparametar koji je zapravo cijena neispravne klasifikacije) je **utežan** za svaki primjerak: $C_i = w(x_i) * C$ (*C* je globalni hiperparametar cijene margine)
  - U ovom slučaju SVM više ne bi zakrivio hiperravninu prema manjinskoj klasi

Primjena ansambala klasifikatora

- Ansambli klasifikatora pokazuju se vrlo korisnima u rješavanju problema nebalansiranosti
- Ansambli su manje osjetljivi od pojedinačnih klasifikatora na nebalansiranost
- U velikom broju slučajeva kombiniraju se s **ponovnim uzorkovanjem** i **učenjem osjetljivim na cijenu**
- Dva pristupa:
  - **Iterativna izgradnja modela**
    - Najčešće algoritmi uzdizanja (engl. (i dalje) *boosting*), ponekad evolucijski algoritmi (ponajviše genetsko programiranje)
  - **Paralelna izgradnja modela**
    - *Bagging*, ponovno uzorkovanje, odabir značajki i smanjenje dimenzionalnosti

Primjer ansambla: RUSBoost

- Primjena slučajnog poduzorkovanja (RUS) i *boosting* algoritma AdaBoost.M2
  - *Boosting* se može smatrati naprednom metodom učenja zasnovanom na cijeni
  - *Boosting* daje iz iteracije u iteraciju veće težine primjercima koji nisu uspješno klasificirani
  - **Motivacija: predstavnici manjinske klase imaju veću šansu da budu neispravno klasificirani te stoga imaju veću šansu da dobiju veću težinu**
  - RUS smanjuje broj primjeraka većinske klase na razinu manjinske klase
  - U kombinaciji s AdaBoostom daje izvrsne rezultate (statistički bolje nego primjena samog AdaBoosta)

C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 1, pp. 185-197, Jan. 2010, doi: 10.1109/TSMCA.2009.2029559.

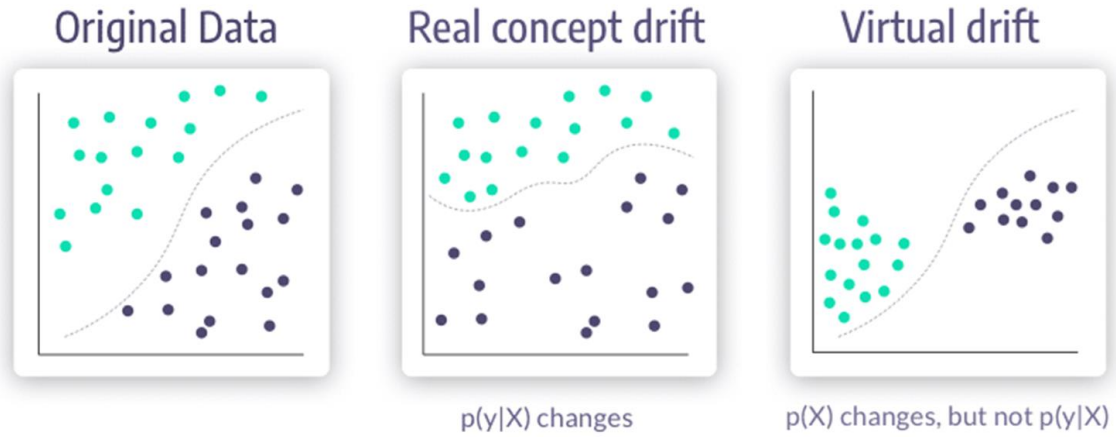
Korištenje prikladnih mjera vrednovanja modela

- **Ako se radi o nebalansiranom skupu podataka**, ukupna klasifikacijska točnost (engl. *total classification accuracy, accuracy*, ACC) **nije dobra mjera** uspješnosti klasifikatora
  - Na temelju te mjere **ne može se ustanoviti koliko je dobro manjinska klasa klasificirana**
- Najčešće je poželjno primijeniti mjere uspješnosti:
  - **Preciznost** (engl. *precision*) i **odziva** (engl. *recall*) – ako želimo procjenu za pojedinačne klase
  - **F1-mjeru** (engl. *F1-score*) – ako želimo kvalitetnu ukupnu procjenu uspješnosti modela
  - I neke druge (npr. *G-mean*)

Pomak koncepta u podacima

Pomak koncepta u podacima

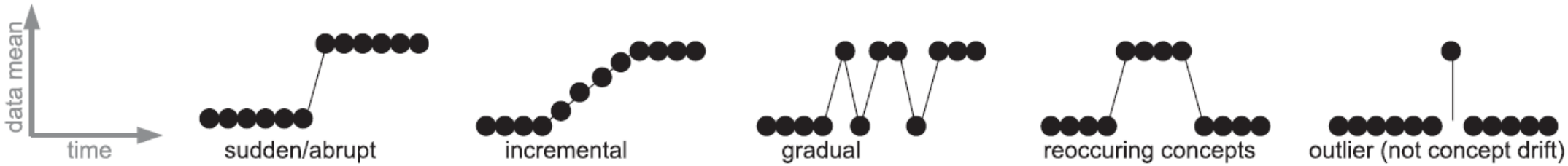
- Pomak koncepta (engl. *concept drift*) u podacima označava pomak (ili odmak) **statističkih svojstava ciljne značajke tijekom vremena**
- Pomak može biti nepredvidiv, a ogleda se u **smanjenju točnosti modela tijekom vremena**
- Matematički izraženo: neka je *x* primjerak podataka, a *w* klasa podatka. Predikcijska funkcija $P_j(x, w) \neq P_k(x, w), j \neq k$, gdje su *j* i *k* različiti vremenski trenutci – optimalnost funkcije u *j* nije zadržana u *k*
- Ako dođe samo do promjene distribucije ulaznih podataka bez promjene (aposteriorne) predikcijske funkcije onda se govori o virtualnom pomaku koncepta, **koji nije problematičan**



Izvor: Micevska S, A Statistical Drift Detection Method, University of Tartu, 2019.

Pomak koncepta u podacima

- Tipovi pomaka u podacima:



Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv (CSUR) 46(4):44

- Nije svaki različit podatak ujedno pomak u podacima, npr. neki su šum (ili stršeći podatak)
- Primjer naglog pomaka: ako na nekoj online usluzi netko tko sluša primarno *pop* glazbu odjednom počne slušati isključivo klasičnu glazbu
- Primjer inkrementalnog pomaka: ako osoba koja sluša *pop* glazbu počne slušati *pop rock* glazbu, a nakon nekog vremena prijeđe na isključivo *rock* glazbu
- **Metode za otkrivanje ovise o tipu pomaka**

Pomak koncepta u podacima

- Primjena detekcije pomaka su detekcija upada u sustav (engl. *intrusion detection*), filtriranje spamova (engl. *spam filtering*), detekcija financijskih prijevara (engl. *fraud detection*), itd.
- U većini ovih problema, **nemamo prethodno znanje o nadolazećim labelama podataka**, a i stvarne labele ne moraju biti odmah poznate
- Kada se dogodi pomak koncepta, **model je potrebno prilagoditi** kako bi se zadržale performanse
- Pristupi detekciji pomaka u podacima
  - **Aktivni monitoring točnosti modela** – tek nakon detekcije promjene ponovno se uči model
  - **Slijepa prilagodba modela** – model se cijelo vrijeme prilagođava s novim podacima bez verifikacije promjena
    - Periodična ponovna izgradnja modela je računski i resursno zahtjevna
  - **Nenadzirano ili polunadzirano učenje za detekciju** – mogu zaobići nedostatke prethodna dva pristupa

Online učenje

- Zajednička značajka metoda koje se predlažu za otkrivanje i karakterizaciju pomaka koncepta
- Algoritmi *online* učenja obrađuju **svaki primjerak za učenje samo jednom**, bez pohranjivanja ili ponovne obrade
- U *online* učenju primjerci dolaze najčešće u obliku **toka podataka** (engl. *data stream*), kada je **fiksan broj značajki**
- Model donosi odluku kada primjerak postane dostupan, što omogućuje sustavu da uči iz pristiglog podatka i revidira dosad naučeni model
- Metode online učenja mogu detektirati pomak **aktivno** (monitoriranje) ili **pasivno** (slijepa prilagodba)

Detektori pomaka zasnovani na monitoriranju

- Detektori pomaka zasnovani na monitoriranju koriste **statističke testove** **da prate je li došlo do promjene u razdiobi ciljne značajke** na temelju performanse pogreške modela
- Smanjenje točnosti je podijeljeno u dvije razine: **upozorenje** i **pomak koncepta**, svaki sa svojim **pragom**
- Po pojavi upozorenja počnu se pamtiti pristigli primjerci, a po pojavi pomaka koncepta **gradi se novi model na temelju starih podataka i prikupljenih zapamćenih primjeraka**, stari model se odbacuje
- Neki poznati algoritmi detektora pomaka zasnovani na monitoriranju:
  - **Metoda otkrivanja pomaka** (engl. *Drift Detection Method*, DDM ili *Statistical Process Control*, SPC)*
  - **Metoda adaptivnog prozora** (engl. *Adaptive Windowing*, ADWIN)
  - Metoda ranog otkrivanja pomaka (engl. *Early Drift Detection Method*, EDDM)**
  - Metoda otkrivanja koja koristi statističko testiranje (engl. *Detection Method Using Statistical Testing*, STEPD)***

*Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv (CSUR) 46(4):44
**Baena-Garcıa M, del Campo-Ávila J, Fidalgo R, Bifet A, Gavalda R, Morales-Bueno R (2006) Early drift detection method. In: Fourth international workshop on knowledge discovery from data streams. vol 6, pp 77–86
***Nishida K, Yamauchi K (2007) Detecting concept drift using statistical testing. In: International conference on discovery science. Springer, pp 264–269

Metoda DDM

- Neka dolaze primjerci (X_i, y_i) za koje neki model SU predviđa klasu $\hat{y}_i$, koja može biti točna predikcija $y_i = \hat{y}_i$ ili lažna $y_i \neq \hat{y}_i$
- Za skup primjeraka, pogreška je slučajna varijabla koja se ravna po binomnoj distribuciji, koja daje vjerojatnost pojave određenog broja pogrešaka u skupu od *n* primjeraka
- Za svaki primjerak, stopa pogreške (engl. *error rate*) je vjerojatnost p_i za opažanje lažne predikcije, sa standardnom devijacijom $\sigma_i = \sqrt{p_i(1-p_i)/i}$
- **Detektor pomaka pamti dvije varijable tijekom monitoriranja modela, p_{min} i σ_{min}**
- **U trenutku *i*, nakon dobivanja predikcije za trenutačni primjerak i ustanovaljavanja stope pogreške, ako je $p_i + \sigma_i$ manji od $p_{min} + \sigma_{min}$ onda $p_{min} = p_i$ i $\sigma_{min} = \sigma_i$**

Metoda DDM

- Za veći broj primjeraka (veći od 30) binomna distribucija približava se normalnoj distribuciji
- Pretpostavka je da se distribucija vjerojatnosti neće promijeniti ako se ne dogodi pomak koncepta
- Interval pouzdanosti od p_i za *n* > 30 primjeraka je približno $p_i \pm \alpha \times \sigma_i$, pri čemu α ovisi o željenom stupnju pouzdanosti
- Kao prag **upozorenja** uzima se pouzdanost od 95%, tj. $p_i + \sigma_i \geq p_{min} + 2 * \sigma_{min}$
- Kao prag **pomaka koncepta** uzima se pouzdanost od 99%, tj. $p_i + \sigma_i \geq p_{min} + 3 * \sigma_{min}$
- Metoda zahtijeva da su labele poznate (nadzirano učenje)

Metoda adaptivnog prozora

- ADWIN koristi prilagodljive prozore ulaznih podataka veličine *W*, a veličina prozora se računa prema stopi promjene koncepta
- Dinamički **povećava veličinu ulaznog prozora** koji razmatra za izgradnju klasifikatora kada nije detektirao pomak u podacima, a **smanjuje veličinu prozora kada je detektirao pomak**
- ADWIN traži dva podprozora od *W* koja imaju **značajno različite prosjeke** određene značajke (prema Hoeffdingovoj granici, vidi kasnije)
  - Ako ih nađe, zaključi da se stariji dio prozora ima različitu distribuciju podataka od trenutne, to se zapamti i taj dio prozora se više neće koristiti za daljnje detekcije
- Dobro se kombinira s uobičajenim klasifikatorima, npr. s *k*-NN-om
  - https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.lazy.KNNADWINClassifier.html

Albert Bifet, Ricard Gavaldà, Learning from Time-Changing Data with Adaptive Windowing. SDM 2007: 443-448

Klasifikatori za učenje iz tokova podataka

- Osnovni zahtjevi za **klasifikatore koji uče na temelju tokova podataka** su mogućnost inkrementalnog poboljšanja, brzina izgradnje modela i osjetljivost na promjene koncepta u podacima (klasifikator ne smije favorizirati stare podatke)
- U teoriji, moguće je koristiti različite algoritme strojnog učenja kao bazne klasifikatore za detekciju pomaka
- U praksi, najčešće se koriste pristupi:
  - Hoeffdingova stabla (engl. *Hoeffding trees*) i **VFDT** pristup
  - Naivni Bayesov algoritam
  - Ansambli klasifikatora (i za nenadzirano i za polunadzirano učenje)

Firas Bayram, Bestoun S. Ahmed, Andreas Kassler, From concept drift to model degradation: An overview on performance-aware drift detectors, Knowledge-Based Systems 245, 2022, p. 108632.

Hoeffdingova stabla

- Stabla odluke koja se grade na **inkrementalan** način tako da uzorci koji najprije dođu sudjeluju u izgradnji prvo korijena stabla, a kasniji uzorci se proslijede u grane prema podjeli u korijenu te se model kontinuirano nadograđuje
- Ideja: **izgradnja novih čvorova nije automatska**, jer se grananje na nekoj značajki događa samo ako je razlika **ΔG** informacijske mjere u određenom čvoru između najinformativnije i druge najinformativnije značajke za grananje **veća od tzv. Hoeffdingove granice** ε (engl. *Hoeffding bound*), neovisne o distribuciji vjerojatnosti primjeraka:
- $\varepsilon = \sqrt{R^2 \ln(\frac{1}{\delta}) / 2n}$, *R* je raspon značajke, *n* je broj primjeraka značajke, 1-δ je vjerojatnost da je prava srednja vrijednost značajke najmanje jednaka $\bar{r} - \varepsilon$, gdje je $\bar{r}$ srednja vrijednost trenutačnog uzorka značajke
- **Čvor postepeno akumulira primjerke** koji pristižu iz toka podataka sve dok se pređe Hoeffdingova granica
- Pokazuje se da se Hoeffdingova stabla izgrade slično uobičajenim stablima odluke kada bi imali odmah na raspolaganju čitav skup podataka

VFDT za Hoeffdingova stabla

- **Vrlo brza stabla odluke** (engl. *Very fast decision trees*, **VFDT**) je klasifikator zasnovan na Hoeffdingovim stablima koji radi određeni broj **optimizacija** Hoeffdingovih stabala kako bi brže i bolje izgradio model
- Ako dvije ili više značajki imaju slične vrijednosti informacijske mjere G onda će trebati puno primjeraka da se odluči po kojoj značajki će se granati te se u tom slučaju grana po onoj koja ima najveću razliku $\Delta \mathbf{G} < \varepsilon < \mathrm{T}$, gdje je T korisnički postavljen prag (*tie threshold*)
- Računanje G traje te da se to ne bi ponavljalo za svaki pristigli primjerak, može se definirati minimalni broj primjeraka koji se treba akumulirati prije novog izračuna G, u oznaci n_{min}
- Ako se ikada postigne ograničenje memorije pri izgradnji stabla, VFDT uklanja najmanje zanimljive listove prema određenom kriteriju
- https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.trees.HoeffdingTreeClassifier.html
- Proširenje: striktna vrlo brza stabla odluke (engl. *strict very fast decision trees*) – jače ograničavaju veličinu stabla

Victor Guilherme Turrisi da Costa, André Carlos Ponce de Leon Ferreira de Carvalho, Sylvio Barbon Junior, Strict Very Fast Decision Tree: A memory conservative algorithm for data stream mining, Pattern Recognition Letters, Volume 116, 2018, Pages 22-28, https://doi.org/10.1016/j.patrec.2018.09.004.

Odabir značajki u tokovima podataka

- **Odabir značajki u tokovima podataka** (engl. *streaming feature selection*)
  - Problem gdje se **broj značajki u skupu podataka mijenja**, tj. nakon određenog vremena dođe nova značajka (broj primjeraka je fiksan)
  - Potencijalno beskonačno dimenzionalan prostor značajki
  - Npr. analiza teksta s Twittera, gdje nove riječi ili skraćenice (tj. značajke) nadolaze tijekom vremena
- Često nije poželjno čekati dok se prikupi velik broj značajki kako bi se napravio odabir značajki
- Značajke treba razmotriti čim se pojave i odlučiti želimo li ih ostaviti u skupu podataka

Odabir značajki u tokovima podataka

- Koraci odabira značajki u tokovima podataka:
  1. **Napučiti** novu značajku iz toka podataka
  2. **Odlučiti** želi li se uključiti novu značajku u skup podataka
  3. **Izmijeniti** skup podataka s nadodanom novom značajkom (s povratkom na 1. korak)
- Veći broj algoritama: Grafting, Alpha-Invest, SAOLA, OSFS, **Fast-OSFS**, OGFS...

AlNuaimi, N., Mehedy Masud, M., Adel Serhani, M., Zaki, N. (2020), “Streaming feature selection algorithms for big data: A survey”, New England Journal of Entrepreneurship. Vol. 18 No. 1/2, pp. 115-137.

Fast-OSFS

- Engl. *Fast Online Streaming Feature Selection*
- Izvorni algoritam OSFS koristi odluku o uključivanju nove značajke u postojeći skup značajki na sljedeći način:
  - Snažno relevantna ili slabo relevantna značajka dodaju se u skup značajki
  - Ako je značajka slabo relevantna, radi se analiza redundantnosti značajke prema Markovljevom prekrivaču
  - Ako se pokaže da je značajka redundantna, uklanja se iz skupa podataka, a ako ne, onda se analiziraju ostale značajke na redundantnost

Fast-OSFS

- Algoritam Fast-OSFS optimira proces redundantnosti tako da mijenja način provjere je li uključivanjem nove značajke neka od dosad uključenih značajki postala redundantna
  - Ne ispituju se svi podskupovi skupa značajki za utvrđivanje Markovljevog prekrivača, nego **samo oni podskupovi koji sadrže novopristiglu značajku**, s postupkom se nastavlja dok se ne postigne ciljana točnost ili broj iteracija (dolaska novih značajki) istekne

X. Wu, K. Yu, W. Ding, H. Wang and X. Zhu, "Online Feature Selection with Streaming Features," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178-1192, May 2013, doi: 10.1109/TPAMI.2012.197.

Zaključak

- Problem nebalansiranosti klasa značajno utječe na rezultate klasifikatora
- Postoje brojne tehnike za ublažavanje tog problema, a najčešće se koristi ponovno uzorkovanje skupa podataka (poduzorkovanje, naduzorkovanje ili oba pristupa)
- Učenje osjetljivo na cijenu i korištenje ansambala klasifikatora poboljšava točnost modela za nebalansirane klase
- Pomak koncepta u podacima može dovesti do problema ako se klasifikator ne prilagodi promjeni
- Danas se koriste različiti postupci za detekciju pomaka u podacima (i analizi novih značajki) i uzimanje pomaka u obzir kod izgradnje modela