Odabir značajki

Dubinska analiza podataka

4. predavanje

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

Ak. god. 2023./2024.

Sadržaj

- Uvod u odabir značajki
- Filterske metode
- Metode omotača
- Ugrađene metode
- Hibridne metode

Uvod u odabir značajki

Odabir značajki

- Postupak **smanjenja dimenzije (broja varijabli)** skupa podataka
- **Zadržava se interpretacija značajki**, jer se one značajke koje se zadržavaju **ne mijenjaju**
- Što uobičajeno želimo postići odabirom značajki:
  - **zadržati rezultat** modeliranja početnog skupa značajki ili ga **poboljšati**
  - **pojednostaviti model** radi boljeg razumijevanja
  - **smanjiti vrijeme** potrebno za izgradnju modela

Primjena

- **Iznimno široka primjena** u čitavom području znanosti o podacima
- Google Scholar upit “feature selection” vraća preko **1,4 milijuna** znanstvenih članaka (usp. “data science” – 1,4 milijun, “data mining” – 4,3 milijuna)
- Najčešća primjena:
  - Područja gdje postoji velik broj značajki i relativno malo primjeraka za učenje (analiza gena, analiza teksta)
  - Kod rješavanja problema gdje se iz sirovih podataka izluči veliki broj potencijalno korisnih značajki (vremenski nizovi, slike, industrijski procesi)
  - Područja gdje je nužno optimirati vrijeme izgradnje i korištenja modela (npr. ugrađeni sustavi, *wearables*, IoT)

Ciljevi

- Specifični ciljevi odabira značajki
  1. Najmanji podskup značajki koji daje **bolje rezultate** (manju pogrešku) nego početni skup
  2. Najmanji podskup značajki koji daje **približno jednake rezultate** kao početni skup značajki
  3. Bilo koji podskup značajki koji daje najbolje rezultate
  4. Rangiranje značajki prema važnosti za zadani cilj
  5. Odabir točno *k* od početnih *M* značajki takvih da daju najbolji rezultat

Složenost problema

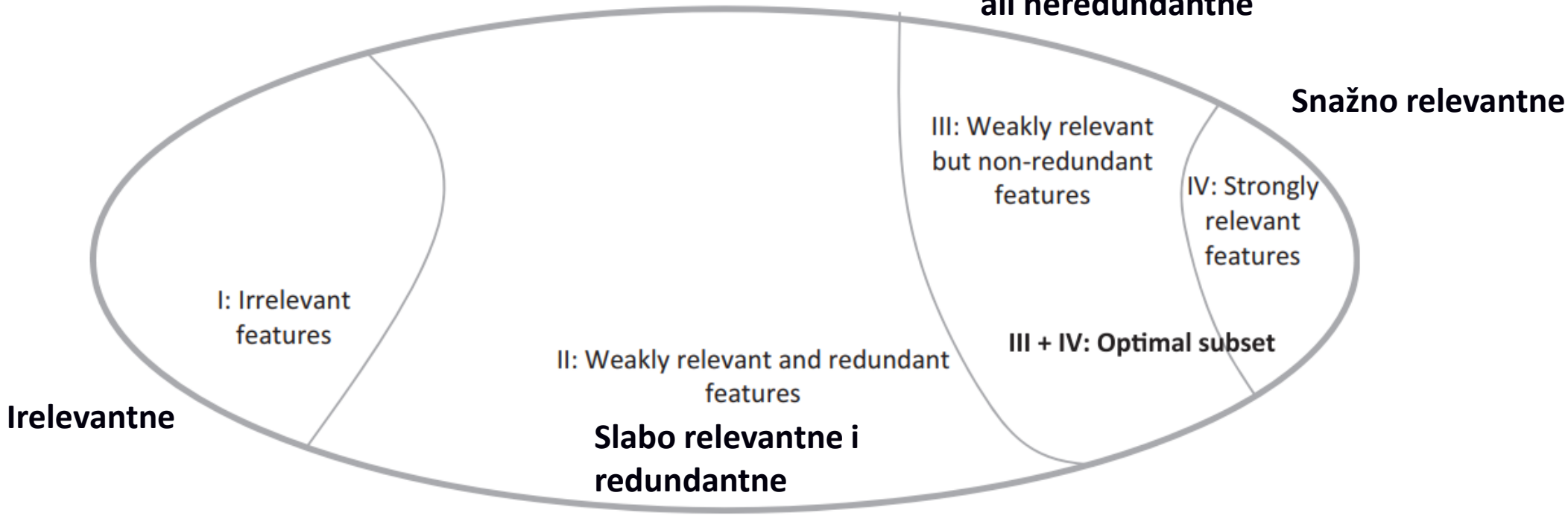
- Iscrpna pretraga za optimalnim podskupom: **NP težak problem**
  - Pretraga 2^M podskupova značajki, gdje je M broj značajki – neizvedivo za veći M (npr. $M > 15$)
- Postojeći empirijski postupci rješavanja obično rade u polinomnom vremenu
  - **Ne garantiraju pronalazak optimalnog podskupa**
  - Obično pronalaze lokalni optimum

Vrste primjene odabira značajki

- Odabir značajki radi se za:
  - **Nadzirano učenje**
    - kriterij je određen s obzirom na odnos vrijednosti značajke prema vrijednosti klase ciljne značajke ili numeričkoj vrijednosti ciljne varijable (za regresijske probleme)
  - **Nenadzirano učenje**
    - kriterij je određen s obzirom na kompaktnost grupa (klastera)
- Prema dostupnosti značajki, razlikuje se:
  - Odabir značajki nad **čitavim skupom** značajki
  - Odabir značajki nad **djelomičnim skupom** značajki (*online* ili *streaming* učenje) tema 5. predavanja

Relevantnost i redundantnost značajki

- Četiri tipa značajki u skupu podataka



Podjela značajki: L. Yu., H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (Oct. 2004) 1205–1224
Izvor slike: N. AlNuaimi, M. Mehedy Masud, M. Adel Serhani, N. Zaki (2020), “Streaming feature selection algorithms for big data: A survey”, New England Journal of Entrepreneurship. Vol. 18 No. 1/2, pp. 115-137.

Terminologija

- Neka je F puni skup značajki skupa podataka dimenzije M, F_i pojedina značajka, C je ciljna značajka, a S_i skup značajki bez F_i: $S_i = F - \{F_i\}$
- Značajka F_i je **snažno relevantna** ako i samo ako vrijedi:
  - $P(C|F_i, S_i) \neq P(C|S_i)$ - uvjetna distribucija vjerojatnosti ciljne značajke C ako su u skupu značajki i F_i i S_i nije ista uvjetnoj distribuciji vjerojatnosti ako je u skupu značajki samo S_i
  - Snažno relevantna značajka je **uvijek bitna** za optimalni podskup značajki
- Značajka F_i je **slabo relevantna** ako i samo ako vrijedi:
  - $P(C|F_i, S_i) = P(C|S_i)$, i $\exists S_i' \subset S_i$, takvi da $P(C|F_i, S_i') \neq P(C|S_i')$
  - Slabo relevantna značajka **nije uvijek bitna** za optimalni podskup, ali u nekim situacijama tijekom smanjenja broja značajki može postati bitna

Terminologija

- Značajka F_i je **irelevantna** (nebitna) ako i samo ako vrijedi:
  - $\forall S_i' \subset S_i \ , \ P(C|F_i, S_i') = P(C|S_i')$
  - Irelevantna značajka **nikada nije bitna** za optimalni skup značajki
- Neka je $M_i \subset F$ takav da $F_i \notin M_i$. M_i se naziva **Markovljev prekrivač** (engl. *Markov blanket*) značajke F_i ako i samo ako vrijedi:
  - $P(F - M_i - \{F_i\}, C|F_i, M_i) = P(F - M_i - \{F_i\}, C|M_i)$
  - Ako razmatramo skup značajki bez Markovljevog prekrivača i značajke F_i te ciljnu klasu, tada je Markovljev prekrivač onaj koji informacijski potpuno prekriva značajku F_i, odnosno distribucija vjerojatnosti u odnosu na ciljnu klasu ne ovisi više o značajki F_i nego samo o Markovljevom prekrivaču

Terminologija

- Može se pokazati da **snažno relevantne značajke nemaju svoj Markovljev prekrivač**
- Neka je *G* trenutačni skup značajki dobiven tijekom procesa odabira značajki (može biti manji od *F*). Značajka F_i je redundantna i može se ukloniti iz skupa *G* ako i samo ako je slabo relevantna i ima neki Markovljev prekrivač M_i unutar *G*
- U praksi, Markovljev pokrivač za odabir značajki ne pronalazi se iscrpno prema definiciji već heuristički (kasnije pokazano)

Podjela metoda odabira značajki

- **Tablične metode** – pretpostavljamo da su značajke neovisne jedna od druge
  - Filterski postupci (engl. *filter methods*)
  - Postupci omotača (engl. *wrapper methods*)
  - Ugrađeni postupci (engl. *embedded methods*)
  - Hibridni postupci (engl. *hybrid methods*)
- **Strukturne metode** – pretpostavljamo da su značajke na neki način povezane jedna s drugom
  - Struktura grafa (engl. *graph structure*), struktura stabla (engl. *tree structure*) ili struktura grupe (engl. *group structure*)

- S. Yang *et al.* Feature Grouping and Selection Over an Undirected Graph. *KDD*. 2012;922-930;
- J. Liu, J. Ye. Moreau-Yosida regularization for grouped tree structure learning, in: Advances in Neural Information Processing Systems, 2010, pp. 1459–1467

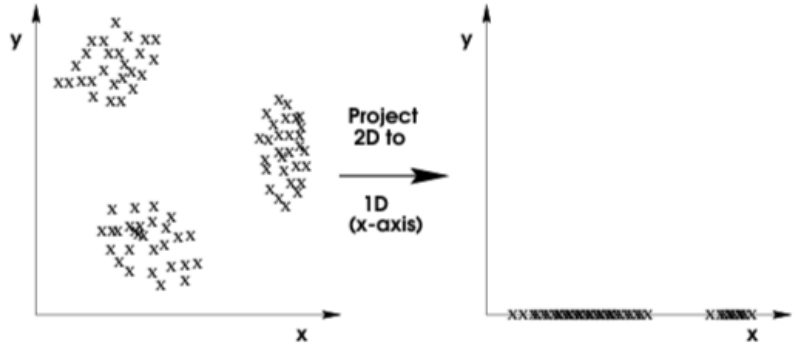
Dva pristupa odabiru značajki

- Određivanje relevantnosti **pojedinačnih značajki** (univarijatni pristup)
  - Rangiraju se prema nekoj mjeri
  - Uklanjaju se ako ne zadovoljavaju neki postavljeni prag
- Određivanje relevantnosti i redundantnosti **podskupa značajki** (multivarijatni pristup)
  - Razmatraju se uvijek podskupovi značajki početnog skupa
  - Računa se značaj podskupa, bilo nekom mjerom bilo rezultatima algoritma za klasifikaciju
  - Pronalazak redundantnih značajki provodi se **implicitno** u postupku

Filterski postupci

Filterski postupci

- Glavna značajka: filterski postupci ne koriste algoritam strojnog učenja da bi napravili odabir značajki
- Filterski postupci definiraju **mjeru** koliko su određena značajka ili skup značajki bitni za opis ciljne značajke
- Razlikuju se filterski postupci koji rade s pojedinačnim značajkama i oni koji rade sa skupovima značajki
  - Za pojedinačne značajke može se odabrati prvih *n* značajki (ili *n*% značajki) za daljnju analizu
- Razlikuju se filterski postupci za nadzirano i za nenadzirano učenje
  - U nastavku razmatramo samo za **nadzirano** učenje



Za nenadzirano učenje vidjeti npr. S. Alelyani, J. Tang, and H. Liu, "Feature Selection for Clustering: A Review," in: C. Aggarwal and C. Reddy (eds.), Data Clustering: Algorithms and Applications, CRC Press, 2013.

Filterski postupci za pojedinačne značajke

- Informacijske mjere
  - **Informacijska dobit** (očekivana zajednička informacija) (engl. *information gain, expected mutual information*)
  - **Simetrična nesigurnost** (engl. *symmetrical uncertainty*)
  - Korelacijski koeficijent (linearni) (engl. *correlation coefficient*) – za regresijske probleme
- Obitelj metoda **Relief**
- hi-kvadrat, χ^2 (engl. *chi-square*, χ^2) – za kategoričke varijable
- Fisherov skor (engl. *Fisher's score*)
- i dr.

- Većina postupaka za odabir značajki implementirana je u Pythonu u paketu:
  https://pypi.org/project/ITMO-FS/

Informacijska dobit

- **Informacijska dobit** za podjelu značajke *X* prema vrijednostima od značajke *Y*:
- IG(*X*|*Y*) = H(*X*) – H(*X*|*Y*), H(*X*) je entropija značajke *X*, H(*X*|*Y*) je uvjetna entropija značajke *X* uz poznavanje *Y*:
  - $H(X) = -\sum_i p(x_i) \log_2 p(x_i)$, $p(x_i)$ je apriorna vjerojatnost za sve vrijednosti *i* od *X*
  - $H(X|Y) = -\sum_j p(x_j) \sum_i p(x_i|y_j) \log_2 p(x_i|y_j)$
- Informacijska dobit je simetrična mjera (poredak *X* i *Y* nije bitan)
  - U praksi, razmatra se odnos između **prediktivne značajke *X*** i **ciljne značajke *Y***
- Vidjeti i: `sklearn.feature_selection.mutual_info_classif`

Simetrična nesigurnost

- Informacijska dobit **favorizira značajke s većim brojem vrijednosti**
- **Simetrična nesigurnost** ograničava vrijednosti informacijske dobiti na interval [0,1]
  - Vrijednost 1 označava potpunu prediktivnost vrijednosti jedne značajke na temelju druge
  - Vrijednost 0 označava neovisnost jedne značajke o drugoj
  - $SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right]$
- SU za odnos između prediktivne značajke i ciljne naziva se još **C-korelacija**, a SU za odnos između dviju prediktivnih značajki naziva se još **F-korelacija**
- U praksi dokazano jako dobra filterska metoda za rangiranje značajki

Obitelj metoda Relief

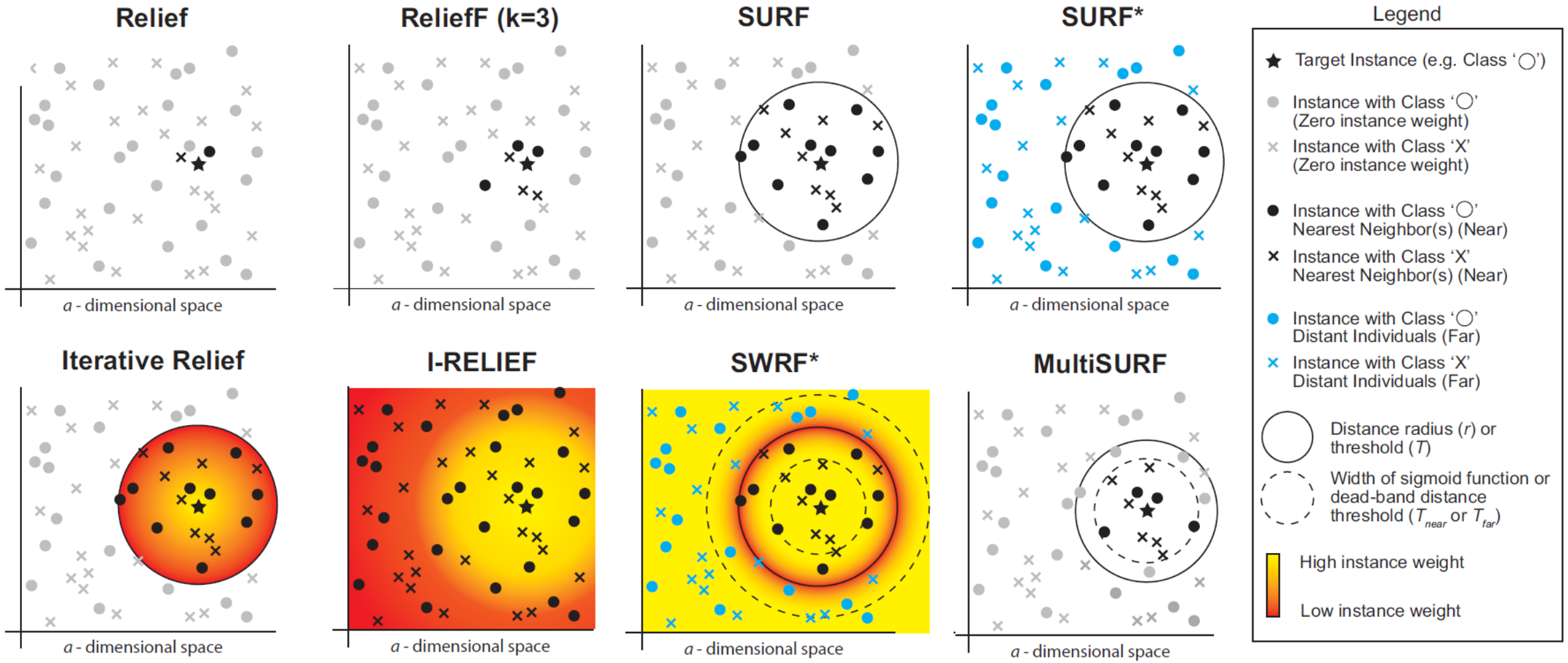
- Veći broj sličnih metoda
- Pojedinačno razmatranje i rangiranje značajki, ali uzimaju u obzir ovisnost među prediktivnim značajkama
- Ne razmatraju podskupove značajki nego su temeljeni na najbližim susjedima za određivanje mjere korisnosti pojedinačne značajke
  - Inspiracija: učenje zasnovano na primjercima (engl. *instance-based learning*)
- Značajke obitelji metoda Relief:
  - nešto **složenije (i sporije)** od ostalih filterskih metoda, ali načelno **točnije**
  - **ne uklanjaju redundantne značajke** iz skupa podataka

Vidjeti: https://gitlab.com/moongoal/sklearn-relief

ReliefF

- Varijanta iskoristiva za **višeklasne** probleme, složenost $\boldsymbol{O}(n^2M)$
- Neka je *n* broj primjeraka u skupu primjera, *M* broj značajki, *k* broj susjeda, a *R* specifični primjerak
- Mjera težine pojedine značajke *W*(*A*) se mijenja iz iteracije u iteraciju prolaskom po svim primjercima i značajkama:
  - $W(A) = W(A) - \sum_{i=1}^{k} \frac{diff(A,R,H_i)}{n \times k} + \sum_{C \neq class(R)} \sum_{i=1}^{k} \left[\frac{P(C)}{1-P(class(R))} \times \frac{diff(A,R,M_i(C))}{n \times k} \right]$
  - Pomoću H_i je označen *i*-ti najbliži pogodak (najbliži primjerak iste klase kao primjerak *R* koji razmatramo, a pomoću M_i je označen *i*-ti najbliži promašaj (najbliži primjerak različite klase)
  - Razlika *diff* se računa ovisno o tome je li značajka kategorička ili numerička
    - Za kategoričku razlika je 0 ako se kategorije značajki podudaraju a 1 inače
    - Za numeričku razlika je: $diff(A, I_1, I_2) = \frac{|value(A,I_1) - value(A,I_2)|}{\max(A) - \min(A)}$

Obitelj metoda Relief



- R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, J. H. Moore, Relief-based feature selection: Introduction and review, Journal of Biomedical Informatics, 85, 2018, pp. 189-203.

Filterski postupci za podskupove značajki

- Najčešći postupci:
  - **mRMR** (engl. *minimum-Redundancy Maximum-Relevance*)
  - **FCBF (engl. *Fast Correlation-Based Filter*)**
  - CFS (engl. *Correlation-based Feature Selection*)
  - Kriterij nekonzistentnosti (engl. *Inconsistency Criterion*)
  - ...

Redom literatura:

H. Peng, F. Long, C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Trans. Pattern Anal. Mach. Intell. 27 (8)(2005) 1226–1238.

L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," in: Proc. 20th International Conference on Machine Learning (ICML-2003), Washington DC, USA, AAAI Press, pp. 856–863, 2003.

M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning". Hamilton, New Zealand, 1998.

H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection-A Filter Solution," in: Proc. 13th International Conference on Machine Learning (ICML-1996), Bary, Italy, Morgan Kaufmann, pp. 319–327, 1996.

mRMR

- Ideja smanjenja utjecaja redundantnih značajki je da se za odabrani skup razmatraju one značajke koje imaju **visoku korelaciju prema ciljnoj značajki i nisku korelaciju prema drugim značajkama**
- Formalno, traži se da **skup značajki** *F* ima maksimalnu relevantnost *D* i minimalnu redundantnost *R*, odnosno da razlika *D* – *R* bude maksimalna, pri čemu:
- $D = \frac{1}{M} \sum_{F_i \in F} IG(F_i|C)$,
- $R = \frac{1}{M^2} \sum_{F_i, F_j \in F} IG(F_i|F_j)$

mRMR

- U praksi, mRMR se provodi inkrementalnim dodavanjem značajke u skup značajki
- Neka već imamo skup F_{m-1} koji se sastoji od nekih *m* - 1 značajki
- Kriterij za dodati *m*-tu značajku (od preostalog skupa od *M* - *m* - 1) u skup značajki je **maksimizacija razlike *D* – *R* za preostale značajke u skupu**, dakle:
- $max_{F_j \in F - F_{m-1}} \left[IG(F_i|C) - \frac{1}{m-1} \sum_{F_i \in F_{m-1}} IG(F_j|F_i) \right]$

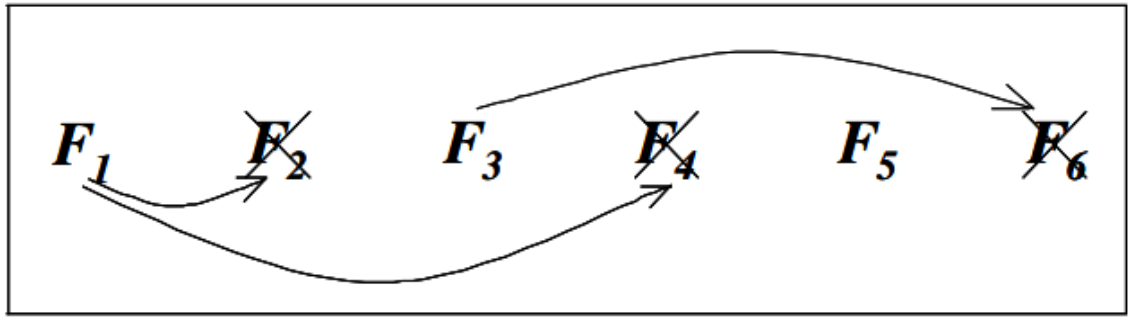
- Mjera mRMR u praksi se pokazala iznimno dobrom za dobivanje relativno **malog skupa (snažno i slabo) relevantnih značajki**

FCBF

- Algoritam zasnovan na mjeri **SU** i **približnom Markovljevom prekrivaču**
- Za dvije relevantne značajke F_i i F_j $(i \neq j)$, F_j formira približni Markovljev prekrivač za F_i ako i samo ako $SU_{j,c} \geq SU_{i,c}$ i $SU_{i,j} \geq SU_{i,c}$
- Intuitivno, želimo zadržati značajku F_j koja ima više informacije o klasi od F_i što se odnosi na dominantnost SU značajke F_j s ciljnom klasom u odnosu na SU značajke F_i s ciljnom klasom
- Dodatno se koristi heuristika C-korelacije $SU_{i,c}$ kao prag da se ustanovi je li F-korelacija $SU_{i,j}$ dovoljno jaka da se značajka F_i ukloni (jer ako značajke nisu jako informacijski bliske onda je teško govoriti o prekrivanju jedne s drugom)

FCBF

- Algoritam radi tako da se najprije rangiraju značajke filterom SU te se odabere prvih *N* značajki koje imaju SU veći od nekog unaprijed definiranog praga δ
- Potom se na preostale značajke, redom po rangu, primjenjuje razmatranje približnog Markovljevog prekrivača prema **svim ostalim** niže rangiranim značajkama
- Preostale značajke čine konačni skup odabranih značajki



- Izvor: L. Yu., H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (Oct. 2004) 1205–1224

Postupci omotača

Postupci omotača

- Glavna značajka: **koriste algoritam strojnog učenja za evaluaciju** određenog podskupa značajki kako bi donijeli odluku o tome je li taj podskup bolji / isti / lošiji od nekog nadskupa
- Algoritam strojnog učenja **često nije onaj** koji se kasnije koristi za izgradnju modela
  - Preferiraju se brzi algoritmi da evaluiraju više skupova značajki – npr. Naivni Bayes, linearni SVM
- U pravilu: **sporiji, ali točniji postupci od filtera**
- **Problem pretrage skupa značajki** (optimizacijski problem) bitan je i za filterske metode, ali je **još bitniji za postupke omotača**, jer one nemaju neku jednostavnu mjeru važnosti nekog podskupa značajki, već moraju graditi model za svaki podskup

Pretraživanje prostora stanja

- Pretraživanje prostora podskupova značajki može početi od **punog skupa** ili od **praznog skupa** i koristiti različite strategije:
  - Slučajno pretraživanje
  - **Pohlepno pretraživanje** (unaprijedna selekcija ili eliminacija unazad)
  - **Slijedna unaprijedna plutajuća selekcija ili eliminacija unazad**
  - Dvosmjerna pretraga (istovremena unaprijed i unazad)
  - Evolucijski algoritmi (engl. *evolutionary algorithms*) – npr. **genetski algoritmi** (engl. *genetic algorithms*)
  - Algoritmi rojeva (engl. *swarm algorithms*), npr. optimizacija rojem čestica (engl. *particle swarm optimization*), optimizacija kolonijom mrava (engl. *ant colony optimization*)

Pohlepno pretraživanje

- Engl. *greedy search ,* sinonimi: slijedna unaprijedna selekcija (*sequential forward selection*) ili slijedna eliminacija unazad (*sequential backward elimination*), uspon na vrh (engl. *hill climbing*)
- **U svakom koraku pretrage dodaje (ili uklanja) jednu značajku koja najviše poveća točnost algoritma strojnog učenja**
- U pretrazi unaprijed ne smanjuje ranije odabrani skup značajki, a u pretrazi unazad ne povećava ranije odabrani skup značajki
- Zaustavlja se izvođenje čim dođe do degradacije točnosti

Slijedna plutajuća selekcija

- Engl. *sequential forward floating selection* (SFFS) / *sequential backward floating selection* (SBFS)
- SFFS (SBFS je obrnut):
  - 1. korak: dodaje jednu značajku u skup koja najviše poveća točnost modela strojnog učenja
  - 2. korak: ukloni jednu značajku iz skupa ako bilo koja od njih pri uklanjanju poveća točnost modela (uklanja se uvijek ona čije uklanjanje dovodi do najvećeg povećanja točnosti)
- Korak 2 se ponavlja dok smanjenje broja značajki povećava točnost i potom se ide na korak 1
- Izvođenje se zaustavlja čim korak 1 ne dovodi do povećanja točnosti
- **Upozorenje**: treba pratiti da ne dođe do beskonačnih petlji s dodavanjima i oduzimanjima značajki

Genetski algoritam za odabir značajki

- Najprije se generira populacija (veličine *N*) na temelju podskupova značajki
- Svaka jedinka **označava jedan mogući podskup** značajki (od ukupno vrlo velikog broja mogućih podskupova)
- Jedinka se obično prikazuje s *M* bitova, gdje je *M* broj značajki, i na početku ima slučajno izabranih *k* značajki postavljenih na 1 (*k* može biti manji od *M*)
- U svakoj iteraciji, sve jedinke se evaluiraju korištenjem modela strojnog učenja
- Primjer uobičajene funkcije dobrote (fitnes funkcije):
  - **fitness = W1 x accuracy + W2 x no_zeros, W1 >> W2** (više vrijedi točnost nego mali broj značajki)

Genetski algoritam za odabir značajki

- Sljedeća generacija sastojat će se od više najboljih jedinki (na turniru) a dodatne jedinke dobivaju se križanjem najboljih jedinki, uz mutaciju (uvođenje ili uklanjanje značajki nasumično)
- Postupak završava nakon određenog broja iteracija
- Preduvjeti uspjeha su:
  - Relativno mali broj značajki *M* u skupu podataka
  - Dovoljno velika populacija
  - Pametan izbor operatora križanja, mutacije i vrste selekcije (način provođenja turnira) za nove jedinke
  - Snažni računski resursi
- Genetski algoritmi obično dobro rade **ako se prethodno iskoriste filterske metode** za eliminaciju velikog broja nebitnih značajki

Ugrađeni postupci

Ugrađeni postupci

- Izbor značajki koji se **temelji na nekom algoritmu strojnog učenja**
- Unutarnja struktura izgrađenog modela oslikava važnost značajki, bilo zbog broja pojavljivanja određene značajke u modelu ili njezine težine (značaja) u modelu
- Mogu se koristiti za dobivanje rangirane važnosti pojedinačnih značajki prema određenom kriteriju ili samo za dobivanje podskupa bitnih značajki
- Primjeri:
  - **Slučajna šuma**
  - Logistička regresija s penalizacijom (LASSO, elastic net)
  - Stroj s potpornim vektorima

Odabir značajki kod slučajne šume

- Slučajna šuma gradi se od **većeg broja stabala odluke** (za detalje vidjeti 6. predavanje)
- U fazi učenja, u svakom čvoru stabla (počevši od korijena) odabire se slučajno jedna značajka po kojoj se grana skup podataka
- Grananje može biti takvo da prouzroči veću ili manju **nečistoću** (engl. *impurity*), pri čemu je grananje čisto (engl. *pure*) ako se njime postigne da u svakom listu postoje samo primjerci jedne klase
- Ideja odabira značajki kod slučajne šume je **uprosječiti u cijeloj šumi koliko svaka značajka smanjuje nečistoću**
  - Značajke koje se nalaze bliže korijenu stabla najčešće više smanjuju nečistoću nego one bliže listovima stabla
- Prednost: brza i jednostavna metoda
- Nedostatci: naglašava važnost numeričkih značajki i kategoričkih s puno vrijednosti, ne uzima u obzir korelaciju značajki – može uključiti kao važne značajke i one međusobno visokokorelirane

Vidjeti: `sklearn.ensemble.RandomForestClassifier.feature_imporatances_`

Hibridni postupci

Hibridni postupci

- Kombiniraju najbolja svojstva filtara i postupaka omotača
- Primjena **dvaju ili više** različitih postupaka filtara, omotača i ugrađenih postupaka
  - Najčešće najprije primijenjen filtar kako bi značajno smanjili prostor značajki
  - Potom primijenjen postupak omotača kojim se nastoji pronaći optimalni podskup značajki
  - Moguće i drugačije kombinacije
- Nema garancije niti da su filtrom zadržane sve bitne značajke niti da se postupkom omotača dobiva najbolji skup
- U praksi se pokazuju **točnijima od filtarskih postupaka** i **bržima od postupaka omotača**

1. primjer hibridnog postupka

- 1. korak: Primijeni ugrađenu metodu (npr. slučajna šuma) i nauči model da bi dobio inicijalni odabrani skup značajki (model treba vratiti određeni skup odabranih značajki), zapamti dobivenu točnost modela
- 2. korak: Primijeni neki postupak filtara (npr. SU) kako bi rangirao odabrani skup značajki
- 3. korak: Ukloni najmanje bitnu značajku po SU-rangu i ponovno nauči model iz 1. koraka na temelju preostalih značajki, zapamti dobivenu točnost
- 4. korak: Provjeri je li uklanjanje značajke iz 3. koraka dovelo do bitnog smanjenja performansi u odnosu na početni model (proizvoljni prag). Ako da, značajka je bitna i ostavi ju u skupu. Nastavi s koracima 3-4 dok se ne isprobaju sve značajke

2. primjer hibridnog postupka

- Prema: Jović et al. 2019 https://www.sciencedirect.com/science/article/abs/pii/S1746809419301636

- 111 ekspertnih značajki varijabilnosti srčanog ritma, klasifikacija subjekata u dvije klase: zdrav ili zatajenje srca
- 1. korak: SU filtar, rangiranje značajki
- 2. korak: Evaluacija podskupova prvih 10%, 20%... 100% rangiranih značajki metodama slučajne šume, SVM-a i drugima, najbolji podskup od 40% značajki zadržan
- 3. korak: Primjena metode omotača s Naivnim Bayesom i SBFS na podskup od 40% značajki, rezultirao s 13 odabranih značajki
- 4. korak: Primjena pohlepne iterativne eliminacije značajki – postupak se obustavlja kada dođe do većeg pada točnosti (procjena) – rezultiralo samo s **4** značajke za klasifikator slučajne šume
- Rezultati: 111 značajki – ACC = 89.8%, 4 značajke – ACC = 90.7%

Zaključak

- Odabir značajki važan je problem u znanosti o podacima i inženjerstvu značajki
- Postoje različiti pristupi koji se razlikuju u brzini i kakvoći pronalaska najboljeg rješenja, pri čemu iscrpno pretraživanje najčešće nije moguće
- Filterski postupci ne razmatraju algoritam strojnog učenja, nego daju internu mjeru značajki
- Postupci omotača vrednuju značajke pomoću rezultata algoritma strojnog učenja
- Ugrađeni postupci vrednuju značajke samom strukturom modela strojnog učenja
- Hibridni postupci kombiniraju gornje pristupe i u praksi često daju jako dobre rezultate