Klasifikacija mina i kamenja iz sonar dataseta

Alen Andrašek, Monika Majstorović, Luka Valenta

1. Opis problema

SONAR (**So**und **N**avigation **a**nd **R**anging) je tehnika koja se koristi za navigaciju, detekciju drugih objekata ili komunikaciju s drugim objektima na ili ispod površine vode pomoću zvuka. Najčešće ju koriste podmornice za detekciju podvodnih mina, drugih podmornica i neprijateljskih brodova te ribari za detekciju jata riba i dubinu mora. Termin sonar odnosi se također na opremu koja generira i prima zvuk pri korištenju tehnike SONAR. Razlikujemo **pasivni** sonar, koji prima zvukove drugih objekata, te **aktivni** sonar koji emitira zvučne impulse te osluškuje njihovu jeku kada se odbiju od objekt.

Mi smo preuzeli skup podataka naziva "Connectionist Bench (Sonar, Mines vs. Rocks) Data Set" (skraćeno, sonar dataset) iz UCI repozitorija za strojno učenje [11]. Podaci su skupljeni tako što su na pjeskovito morsko dno stavljeni metalni cilindar i kamenje otprilike cilindričnog oblika, oboje duljine oko 1.52 m (5 ft) te je pomoću aktivnog sonara emitiran širokopojasni *linearni chirp* (frekvencija pulsa mijenja se linearno s vremenom). Jeka je prikupljena s udaljenosti 10 m te iz kuteva raspona do 90° za cilindar i raspona do 180° za kamen.

Od 1200 primljenih zvukova, odabrano ih je 208 koji su imali omjer šuma i signala između 4 dB i 15 dB. Od tih 208 primljenih signala, njih 111 pripada metalnom cilindru a 97 kamenju.

Daljnjom spektralnom analizom signala te normalizacijom dobivenih podataka svaki od 208 signala predstavljen je konačno 60-dimenzionalnim vektorom u kojem svaka komponenta poprima vrijednosti između 0.0 i 1.0. Svaki od tih 60 brojeva predstavlja energiju unutar određenog frekvencijskog pojasa, integriranu tijekom određenog vremenskog razdoblja.
U ovom skupu podataka nema vrijednosti koje nedostaju.

2. Cilj projekta

Želimo trenirati što bolji model za klasifikaciju sonar dataseta. Konkretno, uspoređivat ćemo performansama SVM-a i k-NN-a međusobno te u usporedbi s performansama u [5].

3. Dosadašnja istraživanja

Sonar dataset često je korišten za razne edukacijsko-demonstrativne svrhe u području strojnog učenja te za benchmark novih modela i algoritama. U ovom poglavlju dat ćemo kratak osvrt na tri istraživanja provedena u zadnjih nekoliko godina u kojima se koristio sonar dataset.

3.1. Gray wolf classifier (GWC)

Trojac s Iranskog sveučilišta znanosti i tehnologije u Teheranu, M.R. Mosavi, M. Khishe i A. Ghamgosar, (2016.) rješavao je problem optimiziranja treniranja neuralnih mreža pomoću algoritma sivog vuka (eng. Gray Wolf Optimization, GWO) [1].

GWO je meta-heuristika inspirirana hijerarhijskom organizacijom čopora i načinom lova sivog vuka. Matematički model za ovo biološko ponašanje predstavljen je 2014. godine (iste godine kada je i rad [1] primljen na recenziju). Od tada se GWO zbog dobrog balansiranja eksploracije (globalna pretraga) i eksploatacije (lokalna pretraga) te lakoće implementacije koristi u raznim poljima.

Klasifikator baziran na algoritmu sivog vuka (GWC) testirali su na tri skupa podataka: Iris, Lenses i Sonar.

Dimenzionalnost Sonar dataseta smanjili su sa 60 na 9 koristeći analizu glavnih komponenti (PCA) zbog jednostavnosti i male računalne kompleksnosti te metode. Odlučili su se na PCA **bez nadzora** jer je ta varijanta pokazala bolje rezultate na stvarnim skupovima podataka.

Rezultate GWC-a uspoređivali su s rezultatima klasifikatora baziranih na algoritmu roja čestica (eng. Particle Swarm Optimization, PSO), algoritmu gravitacijske pretrage (eng. Gravitational Search Algorithm, GSA) te hibridu prethodna dva: PSOGSA. Promatrali su classification rate, convergence rate i sposobnost izbjegavanja zaglavljivanja u lokalnom minimumu.

Zaključili su da GWC daje najbolje rezultate u sva tri promatrana aspekta na sva tri skupa podataka.

3.2. Online multiple kernel klasifikacija (OMKC)

Online učenje metoda je strojnog učenja u kojoj podaci postaju dostupni redoslijedom i koriste se za ažuriranje našeg najboljeg prediktora za buduće podatke u svakom koraku.

U [3] autori R. K. Jade, L. K. Verma i K. Verma klasificiraju sonar dataset koristeći online multiple kernel klasifikaciju.

OMKC koristi online učenje da bi naučila predikciju koja se temelji na kernelima tako što odabire podskup unaprijed definiranih kernela. Algoritam koji su koristili za online učenje je kombinacija perceptron algoritma koji uči klasifikator za dani kernel i hedge algoritma koji kombinira klasifikatore s linearnim težinama.

Razvili su stohastičke strategije odabira koje slučajno odabiru podskup kernela za ažuriranje kombinacije i modela, čime su poboljšali učinkovitost učenja.

Testiranja su provedena na sonar datasetu primjenjujući sljedeće metode:

- Neuralna mreža
- Perceptron
- Perceptron (uniforman)
- Online učenje
- OMKC s deterministčkim ažuriranjem i deterministčkom kombinacijom algoritmama
- OMKC s stohastičkim ažuriranjem i deterministčkom kombinacijom algoritmama
- OMKC s deterministčkim ažuriranjem i stohastičkim kombinacijom algoritmama
- OMKC s stohastičkim ažuriranjem i stohastičkim kombinacijom algoritmama

Neuralnim mrežama dobivena je preciznost od 89% na trening setu i 83% na setu za testiranje. Najmanje greške dobivene su koristeći OMKC s deterministčkim ažuriranjem i determinističkom kombinacijom algoritmama (24.74%) i OMKC sa stohastičkim ažuriranjem i determinističkom kombinacijom algoritmama (24.83%). Sve kombinacije koristeći OMKC imale su grešku ispod 26%.

Ova eksperimentalna studija tako se pokazala kao obećavajuća izvedba predloženih algoritama za OMKC u učinkovitosti učenja i točnosti predviđanja.

3.3 Ansambli neuronskih mreža

U članku [4] proučava se korištenje ansambl tehnike *bagging* (skraćeno od bootstrap aggregating) na klasifikatore dobivene neuronskim mrežama.
Ansambl je skup nezavisno treniranih klasifikatora čija predviđanja se kombiniraju nekim statističkim metodama. Proučava se efekt na sonar i ionosphere datasetu.

Jedna neuronska mreža bi mogla dati nezadovoljavajuće rezultate zbog mnogih razloga. U medicinskim, financijskim i sl. primjenama se zahtjeva najveća točnost, ali treniranjem više mreža i biranjem najbolje, malo je vjerojatno dobivanje znatno veće efikasnosti.

Umjesto toga, moguće je koristiti bagging - istrenirati veći broj klasfikatora te ih spojiti u jedan konačni, najbolji klasifikator. Tako je konačni klasifikator (težinsko) usrednjenje više njih. Bagging je samo jedna od metoda učenja ansamblima.

Klasifikator korišten u članku je BNPP (Back Propagation Neural Network). Istražena je točnost: jedne mreže zasebno, usrednjene tri mreže (*3-bagged* BNPP) i 25 (*25-bagged*) usrednjenih mreža te 25 usrednjenih mreža s PCA (Principle Component Analysis). (Implementirano u Matlab 7). Training set je bio 70% podataka (66 kamena i 80 mina).

Točnosti pojedinih algoritama za Sonar dataset:
- BNPP: 88.709%,
- 3-bagged BNPP: 80.645%,
- 25-bagged BNPP: 93.54%,
- 25-bagged BNPP, PCA: 91.935%.

Može se uočiti kako usrednjavanje malog broja je oštetilo efektivnost klasifikatora, ali usrednjavanjem većeg broja uočavamo bitno poboljšanje. Vidimo da je korištenje PCA nešto smanjilo točnost za ovu metodu i autori zaključuju da se vjerojatno ne isplati koristiti.

4. Plan istraživanja

Sonar dataset je malen pa će biti teže izbjeći overfitting, a outlieri i buka (eng. noise) imat će veći utjecaj nego kod većih datasetova. U slučaju malih datasetova preporučuje se korištenje jednostavnijih modela i modela s manje parametara [6]. Uzimajući u obzir prethodne tvrdnje, odlučili smo koristiti SVM (Suppport Vector Machine) i k-NN (k-Nearest Neighbour).

Podijelit ćemo dataset na 13 disjunktnih podskupova od kojih će svaki podskup imati 16 podataka. Ova podjela neće biti potpuno nasumična. Koristit ćemo stratifikaciju budući da klase u datasetu nisu potpuno balansirane (mina ima više nego kamenja). Tako ćemo osigurati da svaki podskup dobro reprezentira cijeli skup s obzirom na postotak mina i kamenja u njemu. Iako ovaj disbalans nije velik, u slučaju malog dataseta preporučuje se stratifikacija kao mjera osiguranja. [7]

Radit ćemo smanjenje dimenzionalnosti koristeći PCA budući da nam nije bitno da zadržimo interpretabilnost featura.

Za cross-validaciju koristit ćemo nested cross-validaciju.
Unutarnjom cross-validacijom optimizirat ćemo hiperparametre, a vanjskom ćemo odabrati model.
Konkretno:
Ponavljamo dok ne testiramo SVM i k-NN na svih 13 podskupova:
Od 13 podskupova odaberemo jedan testni podskup (koji do sada nije bio testni).
Preostalih 12/13 nekoliko puta nasumičnom stratifikacijom dijelimo u 10 podskupova na kojima vršimo 10-fold cross-validaciju kako bismo optimizirali hiperparametre. Nakon toga, na odabranom test skupu testiramo SVM i k-NN.

Odabrali smo k-fold cross-validaciju jer je prikladnija od hold-out cross-validacije u slučaju malog dataseta [8] jer izostavlja manje podataka pa nam ostaje više podataka za treniranje. Iako se za male datasetove u mnogo slučajeva preporuča leave-one-out cross-validacija, mi smo odabrali izvršiti više puta k-fold cross-validaciju kako bismo smanjili varijancu [12].
Očekujemo da će cross-validacija pomoći kod problema overfittinga.

Pri optimizaciji hiperparametara koristit ćemo random search jer su J. Bergstra i Y. Bengio u [9] utvrdili da random search prostora parametara daje bolje rezultate nego grid search i ima manju računalnu složenost.

Koristit ćemo SVM s kernelom RBF (Gaussian Radial Basis Function). Ovaj kernel često je korišten u primjenama, a odabrali smo ga umjesto linearnog kernela zbog smjernica Andrewa Nga na Coursera Machine Learning kursu (Week 7).

Kod k-NN, pokazano je u [10] da odabir metrike uvelike može utjecati na poboljšanje rezultata na različitim klasifikacijskim problemima. Također, u istoj studiji pokazalo se da k-NN s pravilno odabranom metrikom može dati performans bolji od SVM-a za isti klasifikacijski problem. Mi ćemo promatrati Euklidsku, Manhattan, Čebiševljevu i chi-kvadrat udaljenost.

Programirat ćemo u Python-u koristeći Jupyter Notebook okruženje i standardne Python biblioteke za strojno učenje kao što je scikit-learn.

Za mjeru performansa koristit ćemo classification error kako bismo naše rezultate mogli uspoređivati s [5].

5. Literatura

[1] M.R. Mosavi, M. Khishe, A. Ghamgosar; *Classification of Sonar Dataset Using Neural Network Trained by Grey Wolf Optimization*; Iran University of Science and Technology, Teheran; Iran, 2016.

[2] R.P. Gorman, T. J. Stejnowski; *Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets*; 1987.

[3] R. K. Jade, L. K. Verma, K. Verma; *Classification using Neural Network and Support Vector Machine for Sonar dataset*; International Journal of Computer Trends and Technology; Vol. 4, Issue 2; pg 116-119; 2013.

[4] H. T. Hassan, M. U. Khalid, K. Imran; Intelligent Object and Pattern Recognition using Ensembles in Back Propagation Neural Network; International Journal of Electrical & Computer Sciences IJECS-IJENS; Vol. 10, No. 6; pg 52-59; 2010.

[5] D. Meyer, F. Leisch, K. Hornik; Benchmarking Support Vector Machines; 2002.

[6] https://medium.com/rants-on-machine-learning/what-to-do-with-small-data d253254d1a89 (Zadnje pristupljeno: 30. travnja 2018.)

[6] https://pdfs.semanticscholar.org/d6dc/df86df3ece94c2c5effe205d105c561ed5eb.pdf (Zadnje pristupljeno: 30. travnja 2018.)

[7] https://stats.stackexchange.com/questions/117643/why-use-stratified-cross-validation-why-does-this-not-damage-variance-related-b (Zadnje pristupljeno: 30. travnja 2018.)

[8] http://cs229.stanford.edu/notes/cs229-notes5.pdf (Zadnje pristupljeno: 30. travnja 2018.)

[9] J. Bergstra, Y. Bengio; *Random Search for Hyper-Parameter Optimization;* Journal of Machine Learning Research 13, pg. 281-305; 2012.

[10] Kilian Q. et al., “Distance Metric Learning for Large Margin Nearest Neighbor Classification”, Journal of Machine Learning Research 10, pg. 207-244., 2009.

[11] https://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks) (Zadnje pristupljeno: 30. travnja 2018.)

[12] https://stats.stackexchange.com/questions/61546/optimal-number-of-folds-in-k-fold-cross-validation-is-leave-one-out-cv-always (Zadnje pristupljeno: 30. travnja 2018.)