Java Perceptron
(előző hétről)

Tapasztalatok:

Két ciklus közül az averaged perceptron esetén rendes esetben a külső cikluson kívül kellene átlagolni viszont én kezdetben a külső cikluson belül használtam és úgy jobb eredményeket adott.

UCIDB + SCIKIT-Learn

Két feladatot próbáltam meg és minkettőre ugyanazt a négy modellt próbáltam ki.

A két feladat a balanced scales és az abalone volt.

Először a mérlegeset választottuk, mert a tulajdonságok és a példák száma kicsi volt. Erre az első modell amit ki szerettünk volna róbálni a Perceptron volt. Aztán egy hasonló de nagyobb adathalmazra is ki szerettem volna próbálni, ez lett az Abalone.

A négy modell a Perceptron, Döntési Fa, KNN és OVR/OVA volt.

A tipelésüket rontani nem volt nehéz inkább underfitting irányába overfittingelni ritkán sikerült.

A tippeket javítani penalty-val tudtam a perceptronnál, feltételezem azért mert default = ‘none’

A KNN súlyozásának átállítása ‘uniform’-ról ‘distance’-re a mérlegnél javította a tengericsigáknál rontotta. (Viszont a training result javult -> overfitting?)

A feladathoz tartozó leírás szerint azt gondolom hogy helyes eredményeket kaptam, de bizonytalan voltam, hogy valóban így van-e.

Feladatmegoldás közben lépéseket tettem:

- Ha kellett az adathalmazt szétszedtem training és test data-ra 80-20 arányban
- Az adatok beolvasáskor string típusúak ezeket int/float-tá konvertáltam
- Modellek példányosítása
  Itt a paramétereket lehet tweak-elni
- Azok tanítása fit(x,y)
- Eredmények kiíratása accuracy_score(y_test,y_predict(x))

Felmerülő kérdések:

- Lineáris elválaszthatóság eldöntése

- Különböző típusú attribute-ok / típuskonverzió

- test results nagyon ritkán jobbak mint a training results, lehetséges?(1-2 alkalom és nem az összes modellre) (Perceptron)

- Decision Tree mindig 1.0 training data-ra.

- példányosít -> fit -> score ?

- Ellenőrzés? Elfogadási feltétel?

Eisenstein PDF

4.fejezet

4.1

vélemény: poz,neg,(semleges)

Rövid szövegekre:
From bag of word to bag of bigrams

subjektivitás
álláspont

Ha a szövegben több alany van(enitiy) mindegyikről próbáljuk megállapítani az író viszonyát

A hat alapérzelem felismerése

4.2

egy szó - több jelentés
kontextus fontos

sok a többjelentésű szó, mindegyiknek kellene egy training set

nehézkes címkézett adatokat szerezni ezért vagy unsupervised vagy semi-supervised learning a használatos

4.3

Tokenizáció (részekre bontás? a szó a mondtat tokenje a mondat a bekezdésé), nyelv függő

stemmer,lemmatizer is nyelvfüggő
(Tőszó és jelentés meghatározása)

stopwords,stoplists - olyan szavak elhagyása amelyek gyakran előfordulnak de kis befolyással bírnak

Lehet elég binárisan vizsgálni hogy szerepelt-e a szó mintsem számolni hányszor. (Ha egyszer szerepelt nagy valószínűséggel többször is fog)

6.fejezet

szöveg / szó után következő szó valószínűségének megtippelése

6.1

Mekkora a valószínűsége hogy x szó adott sorrendben lesz?
Nagyon kicsi mert sokféle sorrend létezik/létezhet
Szükségünk lesz bias-ra

n-gram
Mekkora a valószínűsége hogy adott szó következik ha előtte adott szavak (n db) voltak?
Egy szó előtti adott darab szó valószínűségét veszi figyelembe?

n hyperparameter lesz, bias-variance tradeoff

6.2

p(w) = 0 elkerülése, pseudo count-ok bevezetése (Naive Bayes, smoothing)

Probability mass “kölcsönvétele” és újraosztása (Discounting)
Nem kötelező egyenletesen visszaosztani (Backoff)
összetartozó szavak számának csökkentése (ha nincsenek)

Kneser-Ney smoothing:
Az ismeretlen szavaknál a sokoldalúságukat használja segítségül.

6.3

Neural language models

szavak valószínűségének megtippelése conetxtustól függően, ami függ a korábbi szavaktól.
diszkrét tanulási feladat

RNN

ismétlődően updatelni a context vektorokat miközben végigmegyünk a sorozaton

a context vector adott eleme befolyásolva lesz az összes előtte lévő szó által és ettől az elemtől fog függeni a szó

LSTM

memory cell bevezetése, ennek segítségével az információ nagy távokon keresztül is tud terjedni gyengülés nélkül

6.4

intristic evaluation task-neutral nem olyan nehéz

intristic eval metric az a valószínűség amit a model a held-out data-hoz rendel

perplexity - zavaroság
minél kisebb ez az érték annál jobb

6.5
az ismeretlen szavakat jelöljök meg UNK tokennel de érdemes lenne az ismeretlen szavakat valószínűségeit is megkülönböztetni, ezek közül némely megjósolható lehet a nyelv szabályaiból, ha hasonít egy ismert szóra.