Klasifikacija tekstualnih dokumenata

Cilj ove vježbe je upoznavanje sa različitim algoritmima za klasifikaciju tekstualnih dokumenata. Kao primjer zadatka klasifikacije iskoristićemo detekciju spam email poruka. Dakle, potrebno je projektovati binarni klasifikator kod kojeg su klase u koje se uzorci klasifikuju spam i ham. U vježbi će se koristiti dva klasifikatora: k najbližih susjeda (kNN) i metod vektora nosača (SVM).

1 Uputstva za praktični rad

1.1 LIBSVM biblioteka

Kao jedan od klasifikatora u ovoj vježbi biće korišćene i SVM. SVM biblioteku LIBSVM je moguće preuzeti sa `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`. Preuzetu ZIP arhivu je potrebno raspakovati na pogodno mjesto, a zatim u MATLAB-ovu listu putanja (**File** > **Set Path**) dodati putanju do direktorijuma `%LIBSVM%\windows`, gdje je `%LIBSVM%` direktorijum u koji je raspakovana LIBSVM arhiva. Nakon ovoga biblioteka je spremna za korišćenje iz MATLAB-a. Uputstvo za pozivanje funkcija iz LIBSVM biblioteke iz komandne linije nalazi se u fajlu `README` u direktorijumu `%LIBSVM%\`, a uputstvo za pozivanje funkcija iz MATLAB-a u fajlu `README` u direktorijumu `%LIBSVM%\matlab`.

Osnovne funkcije iz LIBSVM biblioteke koje ćemo koristiti u ovoj vježbi su `svmtrain` i `svmpredict`. Pomoću `svmtrain` se obučava SVM klasifikator. Sintaksa funkcije je

```
model = svmtrain(training_label_vector, training_instance_matrix
                 [, 'libsvm_options']);
```

- `training_label_vector` je vektor dimenzija $m \times 1$ koji sadrži oznake primjera iz trening skupa. Tip ovog vektora mora biti `double`.

- `training_instance_matrix` je matrica dimenzija $m \times n$ koja sadrži m instanci trening uzoraka sa po n obilježja. Tip ove matrice mora biti `double`.
- `libsvm_options` je string koji sadrži opcije algoritma za obučavanje kao što je opisano u `%LIBSVM%\README`. Neke od korisnih opcija su:
  - `-t tip_kernela` – izbor tipa kernela (predefinisano 2):
    - 0 – linearni: $u^T v$,
    - 1 – polinomski: $\left(\gamma u^T v + c_0\right)^{red}$,
    - 2 – radijalna bazna funkcija: $\exp\left(-\gamma \|u - v\|^2\right)$
    - 3 – sigmoidni: $\tanh\left(\gamma u^T v + c_0\right)$
    - 4 – unaprijed izračunati kernel (vrijednosti u zadatom fajlu)
  - `-d red` – red kernel funkcije za polinomski kernel (predefinisano 3)
  - `-g gama` – vrijednost γ za RBF kernel (predefinisano 1/broj_obilježja)
  - `-c parametar_C` – vrijednost parametra C

Funkcija `svmtrain` vraća model koji se može koristiti za predikciju klasa kojim pripadaju novi uzorci. Polja ove strukture opisana su u dokumentaciji.

Pomoću funkcije `svmpredict` određuje se predikcija klase kojoj pripada novi uzorak. Sintaksa ove funkcije je

```
[predicted_label, accuracy, decision_values/prob_estimates] =
          svmpredict(testing_label_vector, testing_instance_matrix,
          model [, 'libsvm_options']);
```

- `testing_label_vector` je vektor dimenzija $m \times 1$ koji sadrži oznake testnih primjera. Ako oznake testnih primjera nisu poznate moguće je koristiti proizvoljne vrijednosti. Tip ovog vektora mora biti `double`.
- `testing_instance_matrix` je matrica dimenzija $m \times n$ koja sadrži m testnih primjera sa po n obilježja. Tip ove matrice mora biti `double`.
- `model` je struktura koju je vratio poziv `svmtrain`
- `libsvm_options` je string sa testnim opcijama koje se zadaju u istom formatu kao i kada se funkcija poziva iz komandne linije.

Prvi izlazni argument funkcije, `predicted_label`, je vektor predikcija oznaka klasa testnih primjera. Drugi izlazni argument, `accuracy`, je tačnost klasifikacije. Treći izlazni argument je vrijednost funkcije odlučivanja, odnosno, estimacije vjerovatnoća da uzorak pripada svakoj od klasa (ako se koristi opcija '`-b 1`').

1.2 Formiranje vektorske reprezentacije dokumenata

MATLAB kod za formiranje vektorske reprezentacije i primjeri email poruka dati su na web stranici predmeta `http://dsp.etfbl.net/multimedia/spam.zip`. U direktorijumu `emails` su email poruke podijeljene u trening i test skupove, te manuelno klasifikovane kao spam ili ham. U datim m-fajlovima implementirano je generisanje rječnika i formiranje vektorske reprezentacije za svaki dokument iz kolekcije.

Kod je organizovan na sljedeći način:

- `spam_classifier`
  Glavni program iz kojeg se pozivaju funkcije za generisanje rječnika i formiranje vektorske reprezentacije za svaki dokument iz kolekcije, te funkcije za obučavanje i testiranje klasifikatora. Dio zadatka u ovoj vježbi uključuje izmjene ovog programa.

- `conf = get_conf()`
  Funkcija koja vraća strukturu `conf` u kojoj su zadati parametri klasifikatora;

- `vocab = generate_vocabulary(conf)`
  Generiše rječnik za trening dokumente koji se nalaze na zadatoj putanji. Ulazni argument je struktura sa parametrima klasifikatora. Vraća ćelijski niz koji sadrži termine koji čine rječnik.

- `[training_set, training_C, test_set, test_C] = compute_collection_representation(vocab, conf)`
  Izračunava vektorske reprezentacije za sve dokumente iz kolekcije. Ulazni argumenti su rječnik i struktura sa parametrima klasifikatora. Vraća matrice `training_set` i `test_set` u kojima su redovi vektorske reprezentacije trening i test dokumenata, te vektore `training_C` i `test_C` koji sadrže oznake klasa za svaki trening, odnosno, test dokument. Vektorske reprezentacije dokumenata su normalizovane tako da je njihova L_2 norma jednaka jedinici.

- `vector = compute_document_representation(filepath, vocab, conf)`
  Izračunava vektorsku reprezentaciju za dokument u fajlu na zadatoj putanji. Ulazni argumenti su i rječnik te struktura sa parametrima. Funkcija vraća vektor čiji elementi su brojevi pojavljivanja pojedinih termina iz rječnika.

- `label = apply_knn(training_set, training_C, test_set)`
  Klasifikator na principu najbližih susjeda. Ulazni argumenti su matrica

`training_set` dimenzija $m \times n$ koja sadrži m trening primjera sa po n obilježja, vektor `training_C` dimenzija $m \times 1$ koji sadrži oznake trening primjera i matrica `test_set` dimenzija $l \times n$ koja sadrži l testnih primjera dimenzionalnosti n. Izlazni argument je vektor predikcija oznaka testnih primjera. Ova funkcija je dio zadatka u ovoj vježbi.

- `Cm = conf_mat(ctest_hat, ctest, Nclasses)`
  Izračunava matricu konfuzija za dati vektor predikcija oznaka testnih primjera, `ctest_hat`, vektor tačnih oznaka testnih primjera, `ctest` i broj klasa. Izlaz iz funkcije je matrica konfuzija.

2 Zadatak

1. Upoznati se sa arhitekturom klasifikatora i osnovnim strukturama podataka.

2. Implementirati klasifikator na principu najbližih susjeda (NN) kao funkciju `apply_nn` čija je deklaracija data u fajlu `apply_nn.m`. Kao mjeru sličnosti dokumenata iskoristiti kosinusnu sličnost.

3. Testirati klasifikator email poruka pokretanjem programa `spam_classifier`. Ako ste tačno implementirali klasifikator tačnost za podrazumijevane parametre (minimalna dužina riječi jednaka 1 i minimalan broj pojavljivanja riječi jednak 1) treba da iznosi 77,5%.

4. U programu `spam_classifier` dodati pozive funkcija za obučavanje i klasifikaciju korištenjem mašina sa vektorima nosačima (SVM).

5. Testirati klasifikator. Ako ste tačno uradili prethodni korak tačnost treba da iznosi 90,5%.

6. Testirati klasifikatore za minimalan broj pojavljivanja riječi iz skupa $\{1, 5, 20\}$ i minimalnu dužinu riječi za vrijednosti iz skupa $\{1, 3\}$. Kako se mijenja tačnost klasifikacije? Za koje vrijednosti se dobija najviša tačnost?

7. Testirati oba klasifikatora u slučaju kada se koriste binarna obilježja. Uporediti rezultate sa prethodnim. Varirajte minimalan broj pojavljivanja riječi i minimalnu dužinu riječi kao u prethodnoj tački. Kako se mijenja tačnost klasifikacije? Za koje vrijednosti se dobija najviša tačnost?

8. Testirati klasifikator na primjeru poruka `email_ham.txt` i `email_spam.txt`.