Goran Brajdić

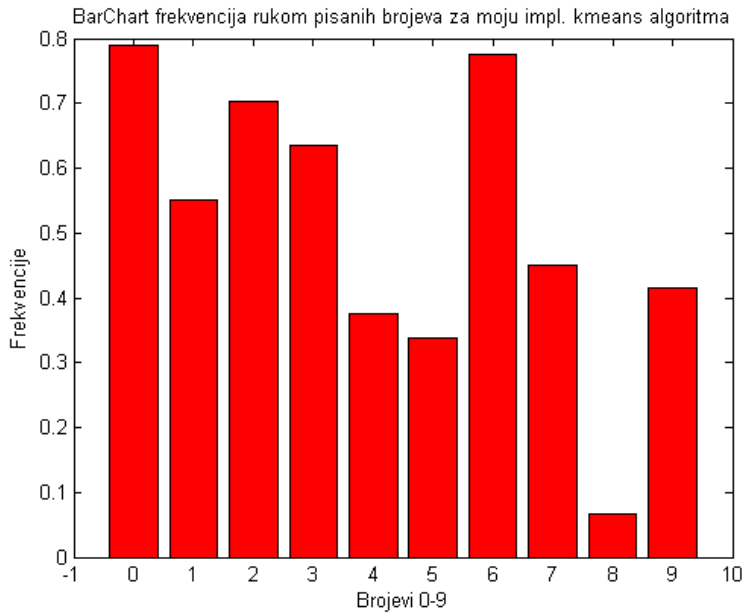
Kvantitativna validacija k-means algoritma za prepoznavanje rukom pisanih brojeva

K-means algoritmom podijelijen je dani trening skup (svih 60 000 testnih primjera) u 10 podskupova (klastera), tada je uzeta srednja vrijednost klastera te iz broja koji je prikazan na slici ručno dodijelijena oznaka 0-9. Nakon toga uzet je skup oznaka (labela testnog skupa) te su dobivene sljedeće frekvencije uspješnosti.

Broj	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Frekvencija	**0.7893**	**0.5512**	**0.7036**	**0.6342**	**0.3745**	**0.3372**	**0.7758**	**0.4506**	**0.0672**	**0.4154**

Iz tablice vidimo da je naš k-means algoritam imao problema sa prepoznavanjem nekih brojeva (*4,5,7,8,9*), dok je ostale brojeve prepoznao relativno dobro. Najlošije je prepoznao broj *8*, jer ga je radi sličnosti svrstavao većinom među brojeve *3,5,6,9*, te je dosta problema bilo i sa *4* i *5*, zbog sličnosti *4* sa *1* te *5* sa *3*.

Sljedeća slika prikazuje stupčasti dijagram frekvencija našeg k-means algoritma.



No kako bismo usporedili koliko je naš algoritam zaista "dobar" moramo ga s nečim usporediti, stoga je cijeli postupak ponovljen sa MatLab-ovim k-means algoritmom.

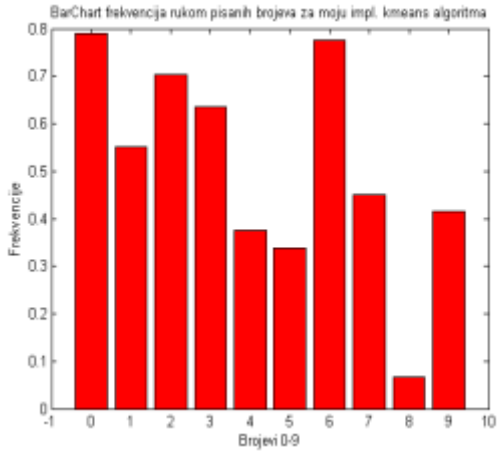
Goran Brajdić

Implementirani k-means algoritam vs MatLab k-means

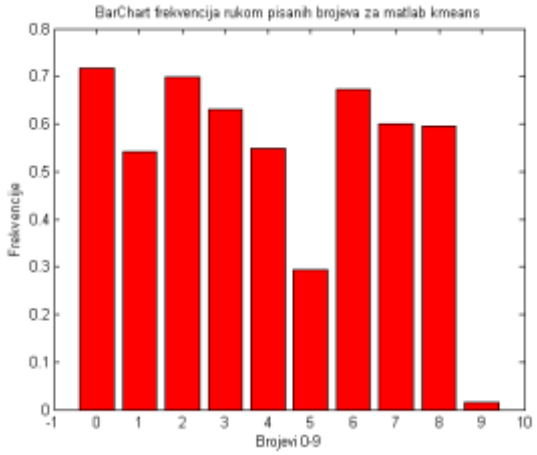
Dobiveni rezultati prikazani su sljedećom tablicom frekvencija uspješnosti.

Broj	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Frekvencija	**0.7172**	**0.5406**	**0.6974**	**0.6309**	**0.5478**	**0.2926**	**0.6735**	**0.6011**	**0.5961**	**0.0145**

Radi usporedbe stupčasti dijagrami frekvencija našeg k-means i MatLab-ovog k-means algoritma dani su redom slikama 1. i 2.



Slika 1.



Slika 2.

Iz slika vidimo da algoritmi daju relativno slične rezultate, razlika je jedino u tome da dok su neki brojevi u našem k-means algoritmu prepoznati bolje, u Matlab-ovom k-means algoritmu su prepoznati lošije i obrnuto. Razlog takvih ponegdje različitih rezultata leži u sličnosti nekih brojeva npr. *8*, *9* i 5 gdje je naš k-means alg. svrstao više brojeva *8* u klastere *9 i 5*, dok je MatLab-ov k-means svrstao više brojeva *9* u klastere *8 i 5.*

Klasifikacija testnih znamenaka algoritmom k-najbližih susjeda

U sljedećem dijelu korišten je algoritam k-najbližih susjeda, i to MatLabov knnclassify algoritam. Skinut je skup testnih primjera (njih 10 000) te je trebalo klasificirati testne znamenke tj. pridjeliti ih nekom od naših klastera dobivenih k-means algoritmom. Algoritam k-najbližih susjeda zapravo uspoređuje svaku znamenku testnog skupa sa znamenkom "training" skupa, tj. u ovoj implementaciji su korištene srednje vrijednosti klastera kao training skup, radi brzine izvođenja samog programa. Algoritam knnclassify je vratio klase, tj indekse 1-10 za svaki od 10 000 znamenaka.

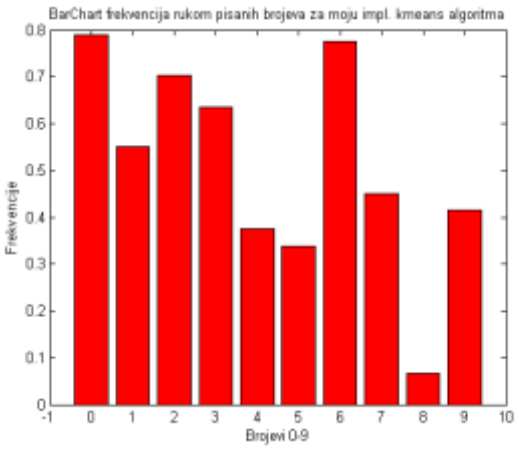
Zatim je skinit skup oznaka (labela) za testni skup, te su napravljene frekvnecije uspješnosti.

Sljedećom tablicom dane su frekvencije uspješnosti za testni skup.

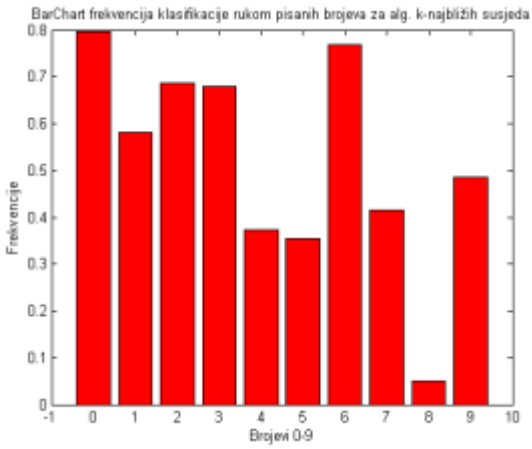
Broj	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Frekvencija	**0.7969**	**0.5815**	**0.6870**	**0.6792**	**0.3727**	**0.3543**	**0.7693**	**0.4144**	**0.0493**	**0.4846**

Iz tablice se vidi da su rezultati vrlo slični rezultatima našeg k-means algoritma, što je potvrda uspješnosti klasifikacijskog algoritma k-najbližih susjeda.

Na sljedećim slikama imamo usporedbu frekvencija uspješnosti k-means algoritma (slika 1), te 10 000 testnih klasificiranih znamenka algoritmom k-najbližih susjeda (slika2).



Slika1.



Slika2.

Dakle i iz stupčastih dijagrama se vidi relativna sličnost među frekvencijama uspješnosti klasteriranja k-means algoritmom, te frekvencija uspješnosti klasificiranih testnih znamenaka.

Analiza frekvencija uspješnosti za različite tipove mjera i normi

U gornjoj analizi u oba algoritma (k-mens i k-najbližih susjeda) korištena je klasična euklidska norma, tj eukl. udaljenost.
U daljnjim analizama usporedit ćemo frekvencije uspješnosti za nekoliko različitih normi.

1. Manhattan norma

Frekvencije uspješnosti k-means algoritma (slika 1), te algoritma k-najbližih susjeda (slika 2)

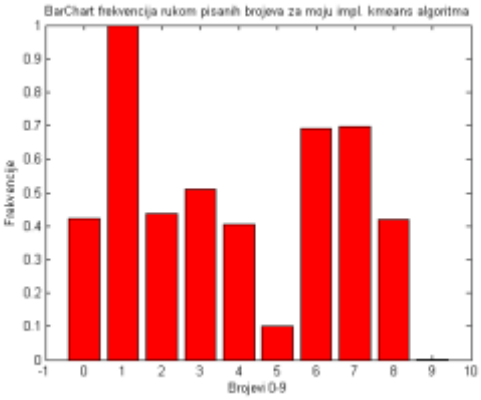
k-means

Broj	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Frekvencija	**0.4214**	**0.9976**	**0.4384**	**0.5100**	**0.4050**	**0.1005**	**0.6914**	**0.6969**	**0.4206**	**0.0024**

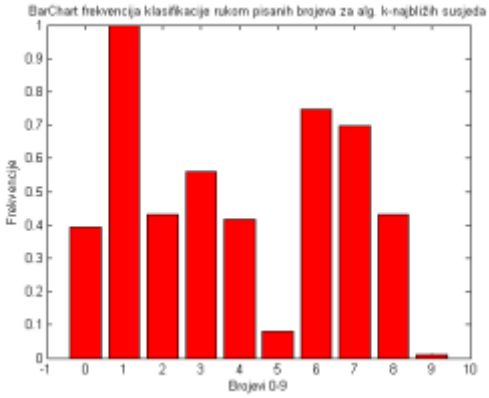
Goran Brajdić

knn

Broj	0	1	2	3	4	5	6	7	8	9
Frekvencija	**0.3929**	**0.9982**	**0.4312**	**0.5604**	**0.4155**	**0.0796**	**0.7463**	**0.6975**	**0.4312**	**0.0089**



Slika 1.



Slika 2.

Vidimo da je k-means alg sa manhattan normom jako dobro prepoznao broj *1 (bolje nego sa eukl. normom)*, te relativno dobro *6* i *7*, no ostale brojeve nešto slabije pogotovo brojeve *5* i *9.*
No generalno obzirom na broj svih prepoznatih brojeva alg. s eukl. normom daje nešto bolje rezultate.

2. Cosine mjera udaljenosti

Računa se kao jedan minus kosinus kuteva točaka (tj vektora, znamenka i centroida).

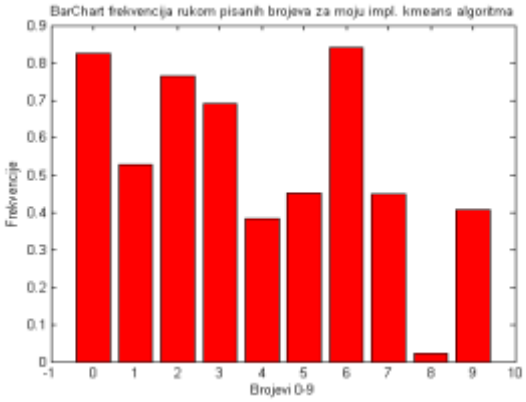
Frekvencije uspješnosti k-means algoritma (slika 1), te algoritma k-najbližih susjeda (slika 2)

k-means

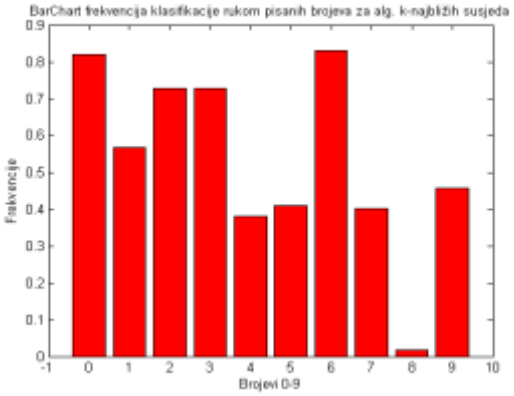
Broj	0	1	2	3	4	5	6	7	8	9
Frekvencija	**0.8261**	**0.5277**	**0.7657**	**0.6917**	**0.3831**	**0.4512**	**0.8417**	**0.4488**	**0.0215**	**0.4071**

knn

Broj	0	1	2	3	4	5	6	7	8	9
Frekvencija	**0.8204**	**0.5683**	**0.7277**	**0.7287**	**0.3798**	**0.4103**	**0.8309**	**0.4018**	**0.0164**	**0.4569**



Slika 1.



Slika 2.

Dakle kod ovakve mjere udaljenosti vidimo da je algoritam vrlo dobro prepoznao brojeve *0, 2, 3 i 6*. Dok je slabije prepoznao brojeve *8, 4 i 7.* U usporedbi sa algoritmom koji koristi euklidsku udaljenost brojevi *0, 2, 3, 4, 5 i 6* imaju veće frekvencije uspješnosti, stoga generalno imamo bolje rezultate nego sa algoritnom koji koristi euklidsku udaljenost.

3. Korelacijska mjera

Računa se kao jedan minus korelacija vektora znamenka i centroida.

Frekvencije uspješnosti k-means algoritma (slika 1), te algoritma k-najbližih susjeda (slika 2)

k-means

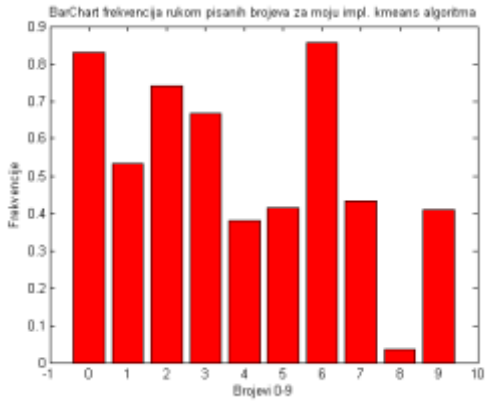
Broj	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Frekvencija	**0.8291**	**0.5316**	**0.7419**	**0.6669**	**0.3795**	**0.4147**	**0.8554**	**0.4330**	**0.0362**	**0.4093**

knn

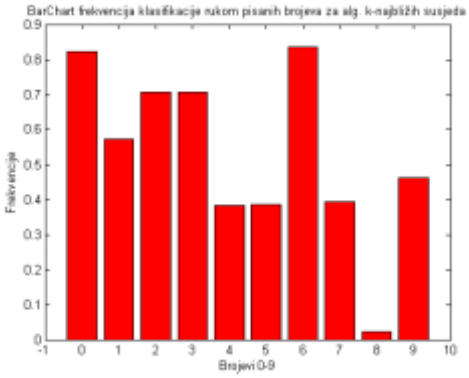
Broj	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Frekvencija	**0.8224**	**0.5718**	**0.7074**	**0.7059**	**0.3819**	**0.3845**	**0.8351**	**0.3930**	**0.0226**	**0.4618**

Frekvencije uspješnsti su vrlo slične kao i kod algoritma sa prethodnom mjerom (jedan minus kosinus kuta). Također vidimo da su vrlo dobro prepoznati brojevi *0, 2, 3 i 6,* dok su slabije prepoznati brojevi *8, 4,i 7.* U usporedbi sa algoritmom koji koristi euklidsku metriku imamo nešto bolje rezultate, za brojeve *0, 2, 3, 5 i 6*.

Goran Brajdić



Slika 1.



Slika2.

Vrijeme izvšavanja algoritma obzirom na karakteristike softvera i stroja

Oba algoritma (k-means te k-najbližih susjeda) implementirani su u Matlabu 7.10.0 (R2010a) 64-bit. K-means je radio kasteriranje na training skupu od 60 000 brojeva, dok je alg. k-najbližih susjeda radio klasificiranje testnog skupa od 10 000 brojeva.
Na računalu Intel Core2 Quad CPU Q8300 @ 2.5GHz 2.5 GHz, 4GB RAM.
Vrijeme izvršavanja programa (mjereno Matlab-ovom funkcijom *tic; toc;)* iznosi 789.886234 sekundi, tj 13.2 minute.