دانشگاه آزاد اسلامی مشهد
دانشکده مهندسی
گروه نرم افزار

شناسایی الگو

گزارش تمرین سوم

استاد محترم:
جناب آقای دکتر معطر

دانشجو:
فاطمه کاکائی ، سمیرا ضیائی

بهمن 94

برای کلاس های تمرین قبل و از هر کلاسی 100نمونه آموزشی تصادفی تولید کنید و از هر کلاس 50نمونه به صورت تصادفی به عنوان نمونه آزمایشی تولید کنید

الف.با استفاده از روش kنزدیک ترین همسایه و با k=5 نمونه آزمایشی را دسته بندی کنید صحت و سرعت یادگیری را مقایسه کنید

ب.با استفاده از روش فشرده سازی condensing، 300نمونه آموزشی را کاهش داده و صحت و سرعت دسته بندی را در روش knn بدست آورید

ج.با استفاده از روش هرس نمونه ها را کاهش داده و صحت و سرعت دسته بندی نمونه های آزمایشی را در این حالت با دو حالت قبل مقایسه کنید

$$\mu 1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma 1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad C1$$

$$\mu 2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \Sigma 2 = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 3 \end{bmatrix} \qquad C2$$

$$\mu 1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad \Sigma 3 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad C3$$

نمونه های آموزشی را برای هر کلاس باتوجه به مقدار میانگین و ماتریس کوواریانس برای هر کلاس با دستور mvnrand (تابع چگالی احتمال توزیع نرمال) بطور تصادفی تولید می نماییم و برچسب کلاس هر نمونه آموزشی را نیز با توجه به آن که توسط کدام کلاس تولید شده است را مقداردهی می نماییم

```
mu1= [0 0];
sigma1= [2 0; 0 1];
numtrainclass1=100;
numtestclass1=50;

mu2= [2 3];
sigma2= [1 0.5; 0.5 3];
numtrainclass2=100;
numtestclass2=50;

mu3= [-2 0];
sigma3= [2 0; 0 1];
numtrainclass3=100;
numtestclass3=50;
```

```
Ntrain=numtrainclass1+numtrainclass2+numtrainclass3;
Ntest=numtestclass1+numtestclass2+numtestclass3;
```

نمونه های تست را برای هر کلاس باتوجه به مقدار میانگین و ماتریس کوواریانس برای هر کلاس با دستور mvnrand (تابع چگالی احتمال توزیع نرمال) بطور تصادفی تولید می نماییم و برچسب کلاس هر نمونه تست را نیز با توجه به آنکه توسط کدام کلاس تولید شده است را مقداردهی می نماییم.

```
%train samples
trainSamples=mvnrnd(mu1,sigma1,numtrainclass1);
trainClass=ones(numtrainclass1,1);
trainSamples=[trainSamples ; mvnrnd(mu2,sigma2,numtrainclass2)];
trainClass=[trainClass; 2*ones(numtrainclass2,1)];
trainSamples=[trainSamples ; mvnrnd(mu3,sigma3,numtrainclass3)];
trainClass=[trainClass; 3*ones(numtrainclass3,1)];
```

```
%test samples
testSamples=mvnrnd(mu1,sigma1,numtestclass1);
testClass=ones(numtestclass1,1);
testSamples=[testSamples ; mvnrnd(mu2,sigma2,numtestclass2)];
testClass=[testClass; 2*ones(numtestclass2,1)];
testSamples=[testSamples ; mvnrnd(mu3,sigma3,numtestclass3)];
testClass=[testClass; 3*ones(numtestclass3,1)];
```

مقدار k=5 را با توجه به صورت سوال مقداردهی می نماییم

الف)

در ابتدا خطی را برای تمایز کردن جواب هر بخش رسم می نماییم و علاوه برآن تعداد نمونه های آموزشی را برای اجرای این قسمت در خروجی نمایش می دهیم.

```
disp('---------- Result of Part1 ----------------')
disp(['number of train samples: ' num2str(Ntrain) ])
```

با دستور زیر زمان شروع برای انجام عملیات دسته بندی نمونه های تست را ثبت می نماییم

```
tic
```

در ابتدا برچسب کلاسی تخمینی تمام نمونه های تست را با صفر مقداردهی می نماییم

```
estimatedTest=zeros(NTest,1);
```

با دستورات زیر برچسب کلاسی تخمینی را برای هر نمونه تست محاسبه و ذخیره می نماییم. حلقه i به تعداد نمونه های تست تکرار می شود و در هر تکرارش برچسب کلاسی یک نمونه را براساس KNN محاسبه می نمایید.

نمونه تست را در متغیر قرار داده

```
for i=1:NTest
  xTest=testSamples(i,:);
  dis=zeros(1,NTrain);
```

بردار فاصله نمونه تست از نمونه های آموزشی را با صفر در ابتدا مقداردهی نمودیم به تعداد کل نمونه های آموزشی اجرا می شود

نمونه آموزشی j را می خوانیم.

```
  for j=1:Ntrain
 xTrain=trainSamples(j,:);
 dis(j)=sqrt(sum((xTest-xTrain).^2));
end
```

فاصله اقلیدسی این نمونه تست i-ام ونمونه آموزشی j-ام را محاسبه نموده وذخیره می نماییم

```
 [~, sortOrder]=sort(dis);

 for j=1:numTrainTotal
    xTrain=trainSamples(j,:);
```

نمونه آموزشی j را می خوانیم.

```
  dis(j)=sqrt(sum((xTest-xTrain).^2));
```

فاصله اقلیدسی این نمونه تست i-ام ونمونه آموزشی j-ام را محاسبه نموده وذخیره می نماییم

```
   end
  [~, sortOrder]=sort(dis);
```

نمونه های آموزشی را به ترتیبی به که به نمونه تست iام نزدیک هستند مرتب می نماییم

```
 KNN=sortOrder(1:K);
```

Kنمونه آموزشی نزدیک داده تست i-ام را تعیین می نماییم

```
 lableKNN=trainClass(KNN);
```

برچسب کلاسی K نمونه نزدیک به نمونه تست i-ام را تعیین می نماییم

```
 estimatedTest(i)=mode(lableKNN);
```

با دستور mode تعیین می نماییم نمونه های کدام کلاس از همه بیشتر در بین K نمونه نزدیک قرار دارند وآن را به عنوان برچسب کلاسی نمونه تست تعیین می نماییم

```
end
```

پس از آنکه برچسب تخمینی را برای هر نمونه تست محاسبه نمودیم بررسی می نماییم که چه تعداد از نمونه های تست را بدرستی تشخیص دادیم و مقدار دقت را با توجه به آن محاسبه می نماییم

```
accuracy=sum(estimatedTest==testClass)/Ntest;
```

زمان اجرا برای انجام عملیات دسته بندی را با دستور زیر محاسبه می نماییم و مقدار آن را نمایش داده

```
toc
```

مقدار دقت دسته بندی بدست آمده برای نمونه های تست را نمایش می دهیم

```
disp(['Accuracy of test Samples= ' num2str(accuracy)])
```

اجرای این بخش به صورت زیر می باشد:

```
. Elapsed time is 0.154095 seconds
Accuracy of test Samples= 0.74          iteration 1

. Elapsed time is 0.450584 seconds
Accuracy of test Samples= 0.83          iteration 2

Elapsed time is 0.455297 seconds .
Accuracy of test Samples= 0.70          iteration 3

Elapsed time is 0.455297 seconds .
Accuracy of test Samples= 0.70          iteration 4
```

ب.ج.ا لگوریتم condensing الگوهای با وزن بالا حذف می شوند اما تاثیر در accuracy ندارد. ایده الگوریتم براساس یک،به اصطلاح زنجیره است. این زنجیره شامل توالی از نزدیکترین همسایه ها به کلاس منتخب است.ما الگوهایی که در پایین زنجیره هستند و نزدیک به مرزهای دسته بندی هستندعلامت گذاری می کنیم و یک راه میان بر برای الگوهایی که ما آنها را به عنوان الگوی آموزشی نگاه می داریم،پیدا می کنیم. این مسئله باعث می شود تعداد نمونه ها به طور موثری کم شوددر حالی دقت دسته بندی با knn تغییر نکند

Condense (train,class,store,trace=true)
Train:matrix for training set
Class:vector of classification for test set
Store:intial store set.default one randomly chosen element of set
Trace:logical.trace iterations

الگوریتم wilson برای کاهش تعداد نمونه های آموزشی بکار می رود. این الگوریتم در ابتدا نمونه های آموزشی را با دسته بند K-نزدیکترین همسایه دسته بندی می نماید (در بعضی مقالات ذکرشده که معمولا تعداد همسایه های 3 می باشد (k=3) و ما نیز از همین مقدار استفاده نمودیم) و سپس بررسی می نماید که برچسب کلاسی کدام یک از نمونه های آموزشی به اشتباه تخمین زده شده است. این الگوریتم نمونه هایی که برچسب کلاسیشان اشتباه تخمین زده شده است را به عنوان نویز درنظر گرفته و از مجموعه نمونه های آموزشی حذف می نماید و بدین ترتیب مجموعه نمونه های آموزشی را اصلاح می نمایید. در پایان در صورتیکه نمونه جدید را بخواهد دسته بندی نمایید از دسته بند نزدیکترین همسایه با تعداد همسایه برابر با یک به همراه مجموعه نمونه های آموزشی اصلاح شده استفاده می نماید و نمونه جدید را دسته بندی می نماید.

در ادامه طیق روال ذکر شده در بالا، الگوریتم را اجرا می نماییم.

در ابتدا برچسب کلاسی تخمینی تمام نمونه های آموزشی را صفر مقداردهی نمودیم

```
estimatedTrain=zeros(numTrainTotal,1);
```

در این مرحله برچسب کلاسی تخمینی را برای تمام نمونه های آموزشی توسط الگوریتم k-نزدیکترین همسایه محاسبه می نماییم که مقدار k در اینجا همانطورکه در توضیحات الگوریتم ذکر شده است برابر با سه می باشد.

```
for i=1:numTrainTotal
   xTest=trainSamples(i,:);
   dis=zeros(1,numTrainTotal);
   for j=1:numTrainTotal
      xTrain=trainSamples(j,:);
      dis(j)=sqrt(sum((xTest-xTrain).^2));
   end
   [~, sortOrder]=sort(dis);
   KNN=sortOrder(2:4);
```

از آنجاییکه در حلقه j فاصله هر نمونه با خودش نیز در بردار dis وجود دارد و پس از مرتب سازی به عنوان نردیکترین نمونه درنظر گرفته می شود پس ما باید آن را درنظر نگیریم که چنین کاری را با درنظر گرفتن از دومین همسایه انجام دادیم

```
   lableKNN=trainClass(KNN);
   estimatedTrain(i)=mode(lableKNN);
end
```

در این مرحله نمونه های آموزشی را که برچسب کلاسیشان توسط knn درست تشخیص داده شده را پیدا می نماییم و براساس آنها مجموعه نمونه های آموزشی اصلاح شده و برچسب کلاسیشان را تعیین می نماییم. بدین ترتیب نمونه های نویزی از مجموعه نمونه های آموزشی حذف می شوند.

```
index=find(estimatedTrain==trainClass);
trainSamplesNew=trainSamples(index,:);
trainClassNew=trainClass(index);
```

تعداد نمونه های آموزشی را بعد از اجرای الگوریتم Wilson محاسبه می نماییم و نمایش می دهیم

```
numTrainTotalNew=numel(trainClassNew);
disp(['number of train samples after of run Wilson: ' num2str(numTrainTotalNew) ])
```

در این قسمت الگوریتم نزدیکترین همشایه را مشابه قسمت اول برای تخمین برچسب کلاسی نمونه های تست اجرا می نماییم و سپس دقت نمونه های تست و مدت زمان اجرا شدن برای انجام دسته بندی را برای تمونه های تست نمایش می دهیم

```
tic
estimatedTest=zeros(numTestTotal,1);
for i=1:numTestTotal
   xTest=testSamples(i,:);
   dis=zeros(1,numTrainTotalNew);
   for j=1:numTrainTotalNew
      xTrain=trainSamplesNew(j,:);
      dis(j)=sqrt(sum((xTest-xTrain).^2));
   end
   [~, sortOrder]=sort(dis);
   KNN=sortOrder(1);
   lableKNN=trainClassNew(KNN);
   estimatedTest(i)=lableKNN;
end
accuracy=sum(estimatedTest==testClass)/numTestTotal;
toc
disp(['Accuracy of test Samples= ' num2str(accuracy)])
.
```

خروجی:

اجرای یک

```
---------- KNN ----------------
number of train samples: 300
Elapsed time is 0.470421 seconds.
Accuracy of test Samples= 0.74
-------- KNN with wilson algorith ----------------
number of train samples after of run Wilson: 234
Elapsed time is 0.350349 seconds.
Accuracy of test Samples= 0.73333
>>
```

اجرا ی دوم

```
---------- KNN ----------------
number of train samples: 300
Elapsed time is 0.454806 seconds.
Accuracy of test Samples= 0.75333
-------- KNN with wilson algorith ----------------
number of train samples after of run Wilson: 212
Elapsed time is 0.308369 seconds.
Accuracy of test Samples= 0.76
```

اجرای سوم:

```
---------- KNN ----------------
number of train samples: 300
Elapsed time is 0.452257 seconds.
Accuracy of test Samples= 0.74667
-------- KNN with wilson algorith ----------------
number of train samples after of run Wilson: 219
Elapsed time is 0.315038 seconds.
Accuracy of test Samples= 0.77333
>>
```

اجرای چهارم:

```
---------- KNN ----------------
number of train samples: 300
Elapsed time is 0.454179 seconds.
Accuracy of test Samples= 0.76667
-------- KNN with wilson algorith ----------------
number of train samples after of run Wilson: 217
Elapsed time is 0.312476 seconds.
Accuracy of test Samples= 0.76
```

با توجه به 4 اجرای بالا مشاهده می نماییم که سرعت اجرای دو روش قسمت ب و ج نسبت به قسمت الف برای دسته بندی نمونه های تست افزایش پیدا کرد (زمان اجرا کاهش پیدا کرد) و علاوه براین تعدادنمونه های آموزشی دو قسمت ب و ج نسبت به حالت اول کمتر شده است و دقت دسته بندی این دو حالت در بعضی موارد بهتر از قسمت الف یا در حد قسمت الف می باشد و در بعضی موارد نسبت به حالت الف کمترشده است که با این حال نزدیک به ان می باشند.