OCR PROJECT

پروژه پایانی

علی رضاقلی زاده

1. چکیده

کاری که در این مقاله انجام شده تشخیص ارقام دست نوشته با طبقه بند های مختلفی چون نزدیک ترین همسایه ، بیز ، پارزن و روش استخراج ویژگی pca است و مطلوب ما مقایسه ی عملکرد این روش ها است . داده ها از 60000 داده ی هُدی که از هر رقم 6000 نمونه وجود دارد انتخاب شده است .

کلمات کلیدی :داده ی آموزش ، داده ی آزمایش ، طبقه بند

2. مقدمه

داده ها شامل 3000 تصویر از ارقام دست نوشته است که به طور دلخواه از هر رقم انتخاب شده است، این تصویر ها با اندازه های مختلف اند که ما آنها را به ماتریس مربعی با سایز ماکزیمم میانگین سطر وستون همه ی تصاویر تبدیل کردیم این سایز برای داده های هدا برابر با 28 است به گونه ای تمام ماتریس تصاویر 28 در 28 شده اند سپس هر تصویر را به یک ماتریس 1×784 تبدیل کرده و بر اساس الگوریتم PCA ، 150 ویژگی از آن انتخاب می کنیم وسپس از آنها 1000 داده را به عنوان آموزش و 2000 داده به عنوان آزمایش به طبقه بند ها می دهیم . ابتدای هر قسمت به تعریفی از آن طبقه بند پرداخته ایم وسپس نتایج اعمال آن طبقه بند را طی جداولی آورده ایم .

3. طبقه بند های مورد استفاده

3.1. نزدیک ترین و k امین نزدیکترین

روش KNN قادر است از میان داده‌های گوناگون که هر یک با یک مجموعه از بردارهای ویژگی مشخص می‌گردند، K داده که به داده مورد بررسی نزدیکترند را انتخاب کرده و سپس با توجه به کلاس در برگیرنده اکثریت داده‌های انتخاب شده، تصمیم نهایی برای طبقه‌بندی بردار مورد بررسی را اتخاذ نماید. مقدار K در این روش همواره عددی انتخاب می شود که منجر به بهترین نتیجه

طبقه‌بندی برای داده‌های آموزش می‌گردد و سپس این مقدار برای طبقه‌بندی داده‌های آزمایش نیز مورد استفاده قرار می‌گیرد.
معیاری که برای سنجش فاصله بین دو بردار در این پروژه به کار گرفته شده است، فاصله اقلیدسی می‌باشد

جدول زیر برچسب خوردن هر یک از داده های آزمایش را با استفاده از این طبقه بند با k=1 نشان می دهد به طور مثال 97٪ از عدد صفر در داده ی آزمایش به درستی تشخیص داده شده است و 2٪ به عدد یک و 0.4٪ به عدد پنج به اشتباه برچسب گذاری شده است :

جدول 1

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
class 1	0.972	0.022	0	0	0	0.004	0	0	0	0
class 2	0.061	0.918	0.010	0	0.005	0	0	0.005	0	0
class 3	0	0.061	0.890	0.042	0.004	0	0	0	0	0
class 4	0	0.005	0.156	0.783	0.021	0	0	0.027	0.005	0
class 5	0.014	0.039	0.099	0.123	0.688	0.009	0	0.009	0.014	0
class 6	0.022	0.039	0	0.004	0.004	0.797	0	0	0.123	0.008
class 7	0	0.020	0.036	0	0	0	0.843	0.072	0	0.026
class 8	0	0.010	0.021	0	0	0	0.015	0.952	0	0
class 9	0.005	0.005	0	0	0	0	0	0	0.978	0.010
class 10	0	0.020	0	0	0	0	0.045	0	0.005	0.928

برای k=2 :

جدول 2

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
class 1	0.99	0.009	0	0	0	0	0	0	0	0
class 2	0.107	0.882	0.005	0	0	0	0	0.005	0	0
class 3	0.009	0.114	0.866	0.004	0.004	0	0	0	0	0

class 4	0.005	0.005	0.254	0.724	0.005	0	0	0.005	0	0
class 5	0.029	0.064	0.183	0.198	0.519	0.004	0	0	0	0
class 6	0.048	0.070	0.004	0.004	0.004	0.779	0	0	0.083	0.004
class 7	0.005	0.041	0.062	0.005	0	0	0.848	0.031	0	0.005
class 8	0	0.010	0.084	0.005	0	0	0.042	0.857	0	0
class 9	0.005	0.016	0	0	0	0	0	0	0.978	0
class 10	0	0.045	0.005	0	0	0	0.081	0	0.005	0.862

جدول 3

Confusion	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
K=1	0.972	0.918	0.890	0.783	0.688	0.797	0.843	0.952	0.978	0.928
K=2	0.99	0.882	0.866	0.724	0.519	0.779	0.848	0.857	0.978	0.862

و میزان درستی (بازدهی) طبقه بند با k=1 ، برابر 0.8745 وبا k=2 ،برابر 0.8310 است که این نشان از برتری نزدیک ترین همسایه است .

3.2. طبقه بند بیز

این طبقه بند بر اساس تحلیل آماری روی داده های آموزش در هر کلاس برای هر داده جدید (داده ی تست) قضاوت می کند بدین گونه که احتمال تعلق داده ی جدیدنسبت به همه ی کلاس ها را بدست می آورد سپس این مقدار برای هر کلاس که بیشتر بود داده جدید را به آن کلاس برچسب گذاری می کنیم .

$$P(w_i \mid x) = \frac{p(x \mid w_i)P(w_i)}{p(x)} \qquad (2\text{-}1)$$

$P(w_i \mid x)$:احتمال تعلق داده x به کلاس w_i است .

$p(x \mid w_i)$: احتمال داده x در کلاس w_i است .

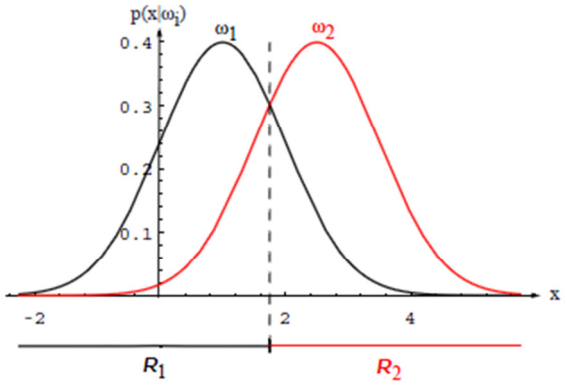
$P(w_i)$: احتمال انتخاب کلاس w_i از میان بقیه ی کلاس ها .

$p(x)$: احنمال قرار گرفتن x در دامنه ی (فضای) کلاس ها .

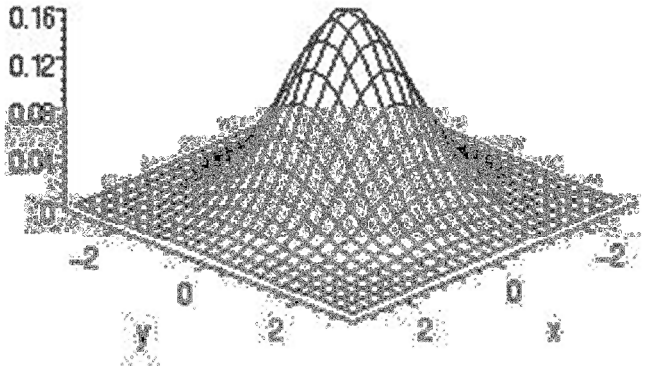
که برای بدست آوردن $p(x \mid w_i)$ لازم است که تابع توزیع آن را داشته باشیم که این مستلزم آن است که ما کل فضای کلاس را داشته باشیم در حالی که ما جز تعدادی داده ی آموزش چیز دیگری از آن کلاس نداریم .

برای این مشکل ، تابع توزیع را تخمین می زنند که ما در اینجا با تابع گوسی این کار را انجام می دهیم .

شکل2 نیز نمونه ای از یک تابع گوسی در دو بعد با میانگین (0, 0) را نشان می دهد .



شکل 1 دو تابع گوسی به عنوان تخمینی برای تابع توزیع دو کلاس w1 و w2 ، نقطه ی x=2 به کلاس w2 برچسب گذاری می شود .



شک2 یک تابع گوسی در دو بعد را نشان می دهد .

با دادن داده های آموزش وآزمایش به این طبقه بند به نتایج زیر می رسیم :

جدول 4

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
class 1	0.915	0.058	0	0	0.001	0.025	0.001	0	0	0
class 2	0.114	0.539	0.001	0	0.13	0.152	0.004	0	0.04	0.02
class 3	0.001	0.003	0.864	0.074	0.039	0	0.015	0.001	0.003	0
class 4	0.001	0.002	0.044	0.899	0.041	0.003	0.002	0.007	0	0.001
class 5	0.002	0	0.001	0.005	0.959	0.033	0	0	0	0
class 6	0	0	0	0	0	0.999	0	0	0.001	0
class 7	0.002	0.001	0	0	0	0.001	0.955	0.003	0.002	0.036
class 8	0.002	0.001	0.001	0	0	0	0.005	0.99	0	0.001
class 9	0	0.003	0	0	0	0.005	0	0	0.992	0
class 10	0	0.001	0	0	0.001	0.003	0.003	0	0.015	0.977

و درصد درستی (بازدهی) آن برابر0.908 است که نشان از برتری نسبی ای به نزدیک ترین همسایه که دارای بازدهی 0.8745 است ، دارد.

جدول 5

Confusion	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
Bayesian	0.915	0.539	0.864	0.899	0.959	0.999	0.955	0.99	0.992	0.977
KNN,k=1	0.972	0.918	0.890	0.783	0.688	0.797	0.843	0.952	0.978	0.928

جدول مقایسه ی بالا نشان از ضعف بیز در تشخیص اعداد صفر و یک و دو و قوت آن در تشخیص بقیه اعداد نسبت نزدیک ترین همسایه است .

3.3. طبقه بند پارزن

روش پارزن روشی ناپارامتری برای تخمین تابع چگالی احتمال است. در این روش که با تعمیم روش هیستوگرام بدست آمده است، به صورت محلی تابع چگالی احتمال را تقریب میزنیم. اساس کار این روش مشابه هیستوگرام است با این تفاوت که ما در این روش خود را مستقیما با داده های با ابعاد بالا درگیر می کنیم. روش کار به صورت زیر است:

گام اول- یک ابر مکعب با طول ضلع h را در نظر بگیرید. حجم این مکعب برابر با $v = h^n$ که n بعد فضای داده ها است.

گام دوم- برای هر نقطه x در تخمین pdf، مرکز این مکعب را بر روی داده مورد نظر قرار می دهیم.

گام سوم- تعداد نمونه های واقع شده در این مکعب را می شمریم. چنانچه این تعداد برابر با K باشد، تخمین ما به صورت زیر در می آید:

$$f(x) = \frac{K}{vN} = \frac{1}{h^n N} \sum_{q=1}^{N} \phi\left(\frac{x - x_q}{h}\right)$$

$$\phi(u) = \begin{cases} 1 & |u| < 0.5 \\ 0 & o.w. \end{cases}$$

که در آن N تعداد نمونه ها است. تابع $\phi(u)$ می تواند مثلثی، گوسی و یا هر تابع دلخواه دیگر باشد، مشروط بر آنکه مساحت زیر آن برابر با یک باشد.

حال نتایج دادن داده ها به این طبقه بند را در جدول زیر آورده شده :

جدول 6

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
class 1	0.918	0.074	0.003	0	0	0	0.004	0	0	0.001
class 2	0.016	0.982	0.001	0	0	0	0.001	0	0	0
class 3	0.001	0.104	0.874	0.018	0	0	0.001	0	0	0.002
class 4	0	0.017	0.152	0.831	0	0	0	0	0	0
class 5	0.001	0.042	0.051	0.031	0.872	0	0.001	0.001	0	0.001
class 6	0.012	0.1	0.012	0.015	0.001	0.856	0.003	0	0	0.001
class 7	0	0.015	0.021	0	0	0	0.955	0	0	0.009

class 8	0	0.005	0.012	0.001	0.001	0	0.014	0.967	0	0
class 9	0	0.094	0.003	0	0	0	0.005	0	0.896	0.002
class 10	0	0.021	0.002	0	0	0	0.008	0.001	0.001	0.967

جدول 7

Confusion	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
Bayesian	0.915	0.539	0.864	0.899	0.959	0.999	0.955	0.99	0.992	0.977
KNN,k=1	0.972	0.918	0.890	0.783	0.688	0.797	0.843	0.952	0.978	0.928
parzen	0.918	0.982	0.874	0.831	0.872	0.856	0.955	0.967	0.896	0.967

نتیجه گیری :جدول فوق نشان از تشخیص بالای طبقه بند پارزن از عدد یک وتشخیص ضعیف بقیه اعداد نسبت به طبقه بند بیز است .جدول فوق این راه کار را به ما پیشنهاد می دهد که برای تشخیص عدد صفر و دو از طبقه بند نزدیک ترین همسایه و برای عدد یک از پارزن و برای بقیه ی اعداد از بیز استفاده کنیم .

Code1. Main

```
load('Data_hoda_full.mat')
% PCA for your database
for i=1:60000
A=imresize(Data{i},[28,28]);
DATA(i,:)=A(:);
end
DATA_arange=PCA(DATA,150);

 [B,IX]=sort(randi(size(DATA_arange,1),[size(DATA_arange,1) 1]),1);
       MixedSet=DATA_arange;
       for l=1:size(DATA_arange,1)
          MixedSet(l,:)=DATA_arange(IX(l),:);
       end
       DATA_arange=MixedSet;
for i=1:10
   j=(DATA_arange(:,end)==i-1);
   DATA2(:,:,i)=DATA_arange(j,1:end-1);
end
DATA_arange2=DATA2;
Xtrain_new=DATA_arange2(1:1000,:,:);
Xtest_new=DATA_arange2(1001:3000,:,:);
[m,n,p]=size(Xtrain_new);
prob_a;
prob_d;
%-----------------------------------------
Xtrain_new=[];
Xtest_new=[];
Xtrain_new=DATA_arange(1:1000,:) ;
Xtest_new=DATA_arange(1001:3000,:);

ClassifiedSet=K_NearestN(Xtrain_new,Xtest_new,1,10);
PerformanceOfKNN=Performance(ClassifiedSet,Xtest_new)
save('result1000')
ClassifiedSet=K_NearestN(Xtrain_new,Xtest_new,2,10);
PerformanceOfKNN=Performance(ClassifiedSet,Xtest_new)
save('2result1000')
ClassifiedSet=K_NearestN(Xtrain_new,Xtest_new,3,10);
PerformanceOfKNN=Performance(ClassifiedSet,Xtest_new)
save('3result1000')
```

Code4. PCA

```
function a=PCA(DATA,d)
DATA=DATA';
load('Data_hoda_full.mat')



DATA_n=double(DATA)-repmat(mean(DATA')',1,60000);
% load DATA_HODA
% PCA
% DATA_n=[1,2,3,4,5;1.5,1.6,3.3,7,9];
C=(DATA_n*DATA_n');
[U D V]=svd(C);
%plot(diag(D)/max(diag(D)))
V_pca=V(:,1:d);
Features=DATA_n'*V_pca;
Features=Features./repmat(max(Features),size(Features,1),1);
DATA_arange=[];
for i=1:10
J=find(labels==i-1);
DATA_arange=[DATA_arange;Features(J,:) repmat(i-1,size(J,1),1)];
end
a=DATA_arange;
save('DATA_arange','DATA_arange')
end
```

Code5 . parzen

```
% load DATA_arange
% Xtrain_new=DATA_arange(1:1000,:,:);
% Xtest_new=DATA_arange(1001:3000,:,:);
% [m,n,p]=size(Xtrain_new);

m=size(Xtrain_new,1);
n=size(Xtrain_new,2);
Xtest_new=Xtrain_new;
mt=size(Xtest_new,1);
%class number
CN=10;

% h=5:5:10;
h=0.5;
performance=zeros(1,size(h,2));
```

```
for T=1:size(h,2)
CCR=zeros(CN,CN);
Conf=zeros(CN,CN);
for i=1:CN
   for j=1:mt
      count=zeros(1,CN);
      for k=1:CN
         a=(Xtrain_new(:,:,k)-repmat(Xtest_new(j,:,i),m,1))<h(T)*ones(m,n);
         count(k)=size(find(sum(a,2)==n),1);

      end
      S=sort(count,'descend');
      [l,C]=max(count);
      Conf(i,C)=Conf(i,C)+(S(1)-S(2))/(S(1)+eps);
      CCR(i,C)=CCR(i,C)+1;
   end
end
performance(T)=trace(CCR)/(mt*CN);
T
performance(T)
Confidence(:,:,T)=Conf;
Confusion(:,:,T)=CCR;
end

figure (1)
plot(h,performance,'--*','LineWidth',2)
set(gca,'fontweight','b')
xlabel('h_s_i_z_e')
ylabel('Fitness')
grid on;

[l,index]=max(performance);
l
ACCR=trace(Confusion(:,:,index))/(mt*CN)
CCR=Confusion(:,:,index)/mt
Conf=Confidence(:,:,index)/mt;
A_conf=diag(Conf(:,1:10))'
A_CCR=diag(CCR(:,1:10))'
```

Code 6: Bayes

```
load DATA_arange

% Xtrain_new=DATA_arange(1:1000,:,:);
% Xtest_new=DATA_arange(1001:3000,:,:);
```

```
% [m,n,p]=size(Xtrain_new);
% n should be the number of features
% m is the number of samples
% p is the number of classes

mu = zeros(p,n);
mu =
[mean(Xtrain_new(:,:,1));mean(Xtrain_new(:,:,2));mean(Xtrain_new(:,:,3));mean(Xtrain_new(:,:,4));mean(Xtrain_new(:,:,5));mean(Xtrain_new(:,:,6));mean(Xtrain_new(:,:,7));mean(Xtrain_new(:,:,8));mean(Xtrain_new(:,:,9));mean(Xtrain_new(:,:,10))];


pdf_mean = zeros(1,n,p);
pdf_var = zeros(n,n,p);

for cnt = 1:p
   hlp=(Xtrain_new(:,:,cnt)-repmat(mu(cnt,:),m,1));
   pdf_var(:,:,cnt) = (Xtrain_new(:,:,cnt)-repmat(mu(cnt,:),m,1))'*(Xtrain_new(:,:,cnt)-repmat(mu(cnt,:),m,1))*(1/(m-1));
end




Conf=zeros(p,p);
CCR=zeros(p,p);

%Making Gaussian Pdf
for k=1:10
for j=1:m
   for i=1:p
      f(j,i)= exp((Xtrain_new(j,:,k)-mu(i,:))*pdf_var(:,:,i)*(Xtrain_new(j,:,k)-mu(i,:))');
   end
end
[a,index]=max(f,[],2);

for l=1:m
   S=sort(f(l,:),'descend');
   [h,index]=max(f(l,:));
   Conf(k,index)=Conf(k,index)+((S(1)-S(2))/(S(1)));
   CCR(k,index)=CCR(k,index)+1;
end

end
Perf=(CCR)/(m);
save('performanceOfBayes','Perf');
```

```
Conf/(m)
ACCR=trace(CCR)/(m*p)
Conf=Conf./(CCR+eps);
CCR=CCR/m;
A_conf=diag(Conf(:,1:p))'
A_CCR=diag(CCR(:,1:p))'
```