
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

# Ejercicios de laboratorio sobre k-vecinos más próximos, PCA y Gaussianas

Alfons Juan

DSIC

Departamento de Sistemas Informáticos y Computación

# Tareas de clasificación

1. ***expressions:*** expresiones faciales representadas mediante vectores 4096-D y clasificadas en 5 clases (1=sorpresa, 2=felicidad, 3=tristeza, 4=angustia y 5=disgusto).
2. ***gauss2D:*** muestras sintéticas procedentes de dos clases equiprobables de forma Gaussiana bidimensional.
3. ***gender:*** expresiones faciales representadas mediante vectores 1280-D y clasificadas por género.
4. ***iris:*** ejemplares de 3 tipos de flores descritos mediante vectores 4-D (longitud y amplitud de pétalos y sépalos).
5. ***news:*** mensajes de grupos de notícias representados mediante bolsas de palabras sobre un vocabulario de talla 100.
6. ***ocr20x20:*** dígitos manuscritos representados mediante vectores 400-D correspondientes a imágenes binarias 20x20.
7. ***videos:*** vídeos de baloncesto o no-baloncesto representados mediante vectores 2000-D extraídos de histogramas de características locales.

# Estadísticas básicas de los conjuntos de datos

| Tarea | $C$ | $D$ | Training | Test |
|---|---|---|---|---|
| expressions | 5 | 4096 | 225 | 48 |
| gauss2D | 2 | 2 | 4000 | 10000 |
| gender | 2 | 1280 | 2836 | 474 |
| iris | 3 | 4 | 18 | 132 |
| news | 20 | 100 | 21701 | 4996 |
| ocr20x20 | 10 | 400 | 1400 | 300 |
| videos | 2 | 2000 | 7985 | 1348 |

Por cada tarea, se proporciona un fichero con datos de entrenamiento, `tareaTr.gz`, y otro con datos de test, `tareaTe.gz`. Se dan en formato de texto `octave`, con una variable `data` de tipo `matrix` que recoge los datos en filas y las características en columnas, salvo la última, que indica la etiqueta de clase (entera).

# Vecino más próximo (NN): error estimado

nn.m

```
#!/usr/bin/octave -qf
if (nargin!=2) printf("%s <tr> <te>\n",program_name()); exit; end
arg_list=argv(); Tr=arg_list{1}; Te=arg_list{2};
load(sprintf(Tr)); tr=data; [NTr,L]=size(tr); D=L-1;
labs=unique(data(:,L)); C=numel(labs);
load(sprintf(Te)); te=data; NTe=rows(te); clear data;
recolabs=zeros(1,NTe);
for i=1:NTe
  tei=te(i,1:D)';
  nmin=1; min=inf;
  for n=1:NTr
    trn=tr(n,1:D)'; aux=tei-trn; d=aux'*aux;
    if (d<min) min=d; nmin=n; endif
  end
  recolabs(i)=tr(nmin,L);
end
[Nerr m]=confus(te(:,L),recolabs);
printf("%s %s %d %d %.1f\n",Tr,Te,Nerr,NTe,100.0*Nerr/NTe);
m
```

```
$ ./nn.m irisTr.gz irisTe.gz
irisTr.gz irisTe.gz 9 132 6.8
m =

   44    0    0
    0   37    7
    0    2   42
```

# PCA y NN: error estimado

eigdec.m

```
function [eigval,eigvec]=eigdec(A)
  [eigvec,eigval]=eig(A);
  [eigval,perm]=sort(-diag(eigval));
  eigvec=eigvec(:,perm);
  eigval=-eigval;
end
```

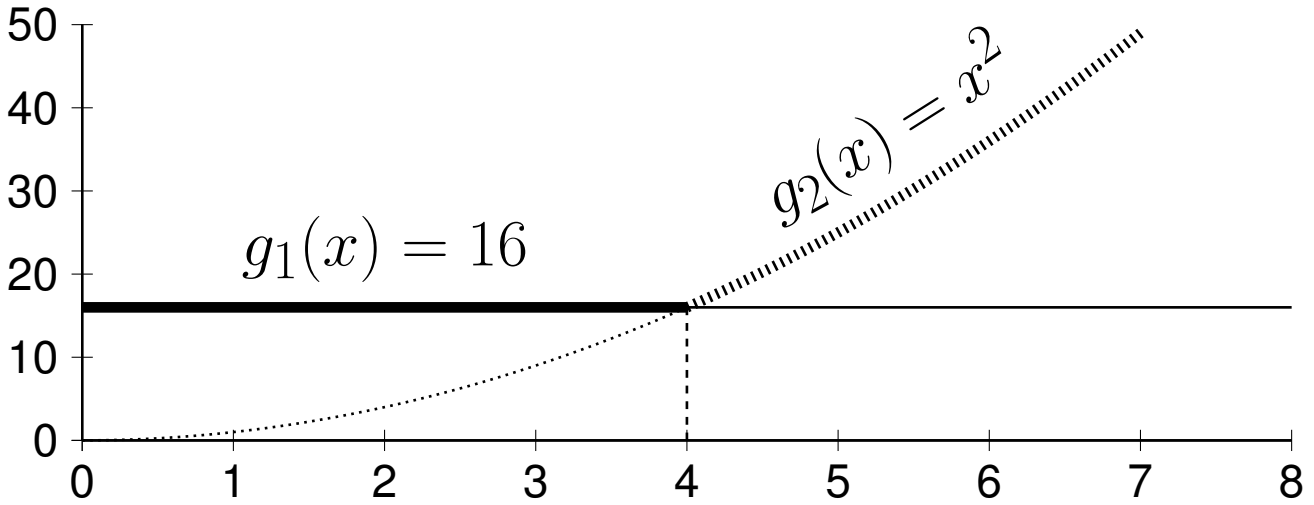
pca_nn.m

```
#!/usr/bin/octave -qf
if (nargin!=2) printf("%s <tr> <te>\n",program_name()); exit; end
arg_list=argv(); Tr=arg_list{1}; Te=arg_list{2};
load(sprintf(Tr)); tr=data; [NTr,L]=size(tr); D=L-1;
labs=unique(data(:,L)); C=numel(labs);
load(sprintf(Te)); te=data; NTe=rows(te); clear data;
S=cov(tr(:,1:D)); [eigval,eigvec]=eigdec(S);
st=sum(eigval); M=1; s=eigval(M);
while (s<.95*st); M=M+1; s+=eigval(M); end;
A=eigvec(:,1:M); trr=tr(:,1:D)*A; ter=te(:,1:D)*A;
recolabs=zeros(1,NTe);
for i=1:NTe
  tei=ter(i,1:M)';
  nmin=1; min=inf;
  for n=1:NTr
    trn=trr(n,1:M)'; aux=tei-trn; d=aux'*aux;
    if (d<min) min=d; nmin=n; endif
  end
  recolabs(i)=tr(nmin,L);
end
[Nerr m]=confus(te(:,L),recolabs);
printf("%d/%d %s %s %d %d %.1f\n",M,D,Tr,Te,Nerr,NTe,100.0*Nerr/NTe);
m
```

# PCA y Gaussiano: error estimado

quadmach.m

```
function cstar=quadmach(W,w,w0,x)
  [D,C]=size(w); cstar=1; max=-inf;
  for c=1:C
    g=x'*W(:,D*(c-1)+1:D*c)*x+w(:,c)'*x+w0(c);
    if (g>max) max=g; cstar=c; endif
  end
endfunction
```




```
W=[0 1]; w=[0 0]; w0=[16 0];
for x=1:8; printf("%d --> %d\n",x,quadmach(W,w,w0,x)); end
```

```
1 --> 1
2 --> 1
3 --> 1
4 --> 1
5 --> 2
6 --> 2
7 --> 2
8 --> 2
```

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

covinv.m

```
function invsigma=covinv(sigma)
  if (issquare(sigma))
    invsigma=pinv(sigma);
  else
    invsigma=1./max(sigma,eps);
  endif
endfunction
```

covlogdet.m

```
function logdet=covlogdet(sigma)
  if (issquare(sigma))
    logdet=log(max(det(sigma),eps));
  else
    logdet=sum(log(max(sigma,eps)));
  endif
endfunction
```

gaussdis.m

```
function [W,w,w0]=gaussdis(prior,mu,sigma)
  [D,C]=size(mu); w0=zeros(1,C); w=zeros(D,C); W=zeros(D,D*C);
  for c=1:C
    muc=mu(:,c);
    sigmac=sigma(:,D*(c-1)+1:D*c); isigmac=covinv(sigmac);
    W(:,D*(c-1)+[1:D])=-0.5*isigmac; w(:,c)=isigmac*muc;
    w0(c)=log(prior(c))-0.5*covlogdet(sigmac)-0.5*muc'*isigmac*muc;
  end
endfunction
```

gaussmle.m

```
function [prior,mu,sigma]=gaussmle(data)
  L=columns(data); D=L-1; labs=unique(data(:,L)); C=numel(labs);
  prior=zeros(1,C); mu=zeros(D,C); sigma=zeros(D,D*C);
  for c=1:C
    datac=data(find(data(:,L)==labs(c)),1:D);
    prior(c)=rows(datac); mu(:,c)=mean(datac)';
    sigma(:,D*(c-1)+[1:D])=(prior(c)-1)/prior(c)*cov(datac);
  end
  prior/=sum(prior);
endfunction
```

pca_gauss.m

```
#!/usr/bin/octave -qf
if (nargin!=2) printf("%s <tr> <te>\n",program_name()); exit; end
arg_list=argv(); Tr=arg_list{1}; Te=arg_list{2};
load(sprintf(Tr)); tr=data; [NTr,L]=size(tr); D=L-1;
labs=unique(data(:,L)); C=numel(labs);
load(sprintf(Te)); te=data; NTe=rows(te); clear data;
S=cov(tr(:,1:D)); [eigval,eigvec]=eigdec(S);
st=sum(eigval); M=1; s=eigval(M);
while (s<.95*st); M=M+1; s+=eigval(M); end;
A=eigvec(:,1:M); trr=tr(:,1:D)*A; ter=te(:,1:D)*A;
[prior,mu,sigma]=gaussmle([trr tr(:,L)]); I=eye(M); a=0.9;
for c=1:C
  sigma(:,M*(c-1)+[1:M])=a*sigma(:,M*(c-1)+[1:M])+(1-a)*I;
end
[W,w,w0]=gaussdis(prior,mu,sigma); recolabs=zeros(1,NTe);
for i=1:NTe
  tei=ter(i,1:M)'; c=quadmach(W,w,w0,tei); recolabs(i)=labs(c);
end
[Nerr m]=confus(te(:,L),recolabs);
printf("%d/%d %s %s %d %d %.1f\n",M,D,Tr,Te,Nerr,NTe,100.0*Nerr/NTe);
m
```

# Actividad

Completa la siguiente tabla de estimaciones de error, mediante partición (*hold-out*), de NN, PCA-NN y PCA-Gaussiano.

| Tarea | NN | PCA-NN | PCA-Gauss |
|---|---|---|---|
| expressions | | | |
| gauss2D | | | |
| gender | | | |
| iris | 6.8 | 11.4 | 5.3 |
| news | | | |
| ocr20x20 | | | |
| videos | | | |

***A entregar:*** un fichero `zip` o `tgz` con:

- La tabla de la actividad completada.
- Opcionalmente, experimentos adicionales que hayan conducido a mejoras en alguna entrada de la tabla.