TPkppv

```
load data_app
xa=x;
Sa=S;
% set d'aprentissage
load data_test
% X et S de data_test -> set de test
xkppv=x;
Skppv = ones(length(S),1);
Sproto = Skppv;

%image
m=300;
im = reshape(x(m,:),28,28)';
figure(1)
image(255*ones(28,28)-im);
colormap(gray)
S(m)
taille_ap = length(Sa)
taille_test = length(S)

% classe 1:10 k = nombre de classe
% classe equirepartie?
k = 10;
for i=1:k
  Ti = mean(Sa==i)
endfor
% oui

%%algo des kppv
% k classe , q voisin
[n c]= size(x);
[l _] = size(xa);
for q = 1:5
 for i  = 1:n


  Temp1 = zeros(1,l);
  for z = 1:l
    Temp1(z) = (xa(z,:)-x(i,:))*(xa(z,:)-x(i,:))';
  endfor
  %%recherche des k ppv pour distance euclidienne

  % Temp1 (j) contient distance au carr entre  xa(j,:) et  x(i,:)
  [_,indice] = sort(Temp1);


  % sort trie par ordre croissant  met dans d les valeur et dans indice les indices
 % correspondants


  % selection des q voisin les plus proche
  selectkvois = indice(1:q);

  % selection des classe auxquelle les q ppv appartienne
  Clkppv = Sa(selectkvois);

  %calcul frequence classe des q ppv
  temp3 = zeros(1,k);
  for j=1:k
    temp3(j) = mean(Clkppv==j);
  endfor

  % affectation ( classe la plus frequente parmis les q ppv )
  [_ Cl] = max(temp3);
  Skppv(i) = Cl;
 endfor

% calcul erreur
erreur = sum(Skppv~=S);
taux_erreur = erreur/length(S)

% matrice de confusion
M_conf = zeros(k);
for ii = 1:k
  %element de classe i
  indclassei = find(S==ii);
  for jj = 1:k
    % element de classe i classer en j
    classerj = find(Skppv(indclassei)==jj);
    %coef i j de la matrice de confusion
    M_conf(ii,jj) = length(classerj);
  endfor
```

```
endfor
q=q
M_conf
endfor


% il y a 4 donnee de classe 6 qui ont ete classee 1 ont les affichent
%image de qq confusion

%comparaison avec un bien classer
ind1 = find(S==6);
class1 = find(Skppv(ind1) == 1);
xprint = x(ind1,:);
for i = 1:length(class1)
  colormap(gray)
  figure(i+10)
  im1 = reshape(xprint(class1(i),:),28,28)';
  image(255*ones(28,28)-im1);
endfor
% le 1 est reconnaissable par un trait 'long' et
%les six classe 1 ont cette caracteristique

% construction des prototype ( 2*moyenne)
Proto = zeros(k,c);
for i = 1:k
  indcli = find(Sa==i);
  Proto(i,:) = mean(2*xa(indcli,:));
  colormap(gray)
  figure(i+10)
  im1 = reshape(Proto(i,:),28,28)';
  image(255*ones(28,28)-im1);
endfor



for i  = 1:n

  Temp1 = zeros(1,l);
  % calcul distance au carr entre Prototype de classe z et x(i,:)
  for z = 1:k
    Temp1(z) = (Proto(z,:)-x(i,:))*(Proto(z,:)-x(i,:))';
  endfor
  % Temp1 (j) contient distance au carr entre  Proto(j,:) et  x(i,:)
  %%recherche du ppv pour distance euclidienne
  [_,indice] = sort(Temp1);
  clppv = Sa(indice(1))
  Sproto(i) = clppv;
endfor

% calcul erreur
erreur = sum(Sproto~=S);
taux_erreur = erreur/length(S);

% il y a surement une erreur car le prototype de type 3 est toujours le plus proche
```

```
ans =  2
taille_ap =  1000
taille_test =  300
Ti =  0.11600
Ti =  0.099000
Ti =  0.093000
Ti =  0.10500
Ti =  0.092000
Ti =  0.094000
Ti =  0.11700
Ti =  0.087000
Ti =  0.10000
Ti =  0.097000
taux_erreur =  0.15333
q =  1
M_conf =
   38    0    0    0    0    0    0    0    0    0
    2   23    0    1    0    0    1    1    0    0
    0    3   17    0    5    0    0    1    2    0
    1    0    0   25    0    2    0    1    9    0
    0    0    2    0   18    1    0    1    1    1
    0    0    0    0    0   25    0    0    0    0
    0    0    0    0    2    0   31    0    1    0
    1    0    1    1    0    1    0   26    1    0
    0    0    0    2    0    0    1    0   29    0
    0    0    0    0    0    0    0    0    0   22
taux_erreur =  0.18667
q =  2
M_conf =
   38    0    0    0    0    0    0    0    0    0
    2   23    0    1    1    0    1    0    0    0
```

```
     0     4    17     0     4     0     1     0     2     0
     3     2     0    30     0     0     0     0     3     0
     0     0     4     0    19     1     0     0     0     0
     3     0     0     0     0    22     0     0     0     0
     0     0     0     2     1     0    30     0     1     0
     2     1     3     1     2     1     0    20     1     0
     0     0     0     2     0     0     2     1    27     0
     0     0     0     0     3     1     0     0     0    18
taux_erreur =  0.18333
q =  3
M_conf =
    38     0     0     0     0     0     0     0     0     0
     2    21     0     0     1     0     1     0     1     2
     0     4    15     0     7     0     0     0     2     0
     3     1     0    27     0     2     0     0     5     0
     0     0     3     0    20     1     0     0     0     0
     2     0     0     0     0    23     0     0     0     0
     0     0     1     3     1     0    28     0     1     0
     1     1     1     0     1     1     0    24     2     0
     0     0     1     0     0     0     1     0    30     0
     0     0     0     0     3     0     0     0     0    19
taux_erreur =  0.17667
q =  4
M_conf =
    38     0     0     0     0     0     0     0     0     0
     3    22     0     1     0     0     1     0     0     1
     0     1    18     0     6     0     0     1     2     0
     2     1     0    29     0     2     0     0     4     0
     0     0     3     1    18     1     0     0     0     1
     4     0     0     0     0    21     0     0     0     0
     0     0     0     2     2     0    29     0     1     0
     1     1     2     0     1     1     0    24     1     0
     0     0     1     1     0     0     1     0    29     0
     0     0     1     0     2     0     0     0     0    19
taux_erreur =  0.17000
q =  5
M_conf =
    38     0     0     0     0     0     0     0     0     0
     2    23     0     0     1     0     1     0     1     0
     0     2    17     0     6     0     0     1     2     0
     3     1     0    29     0     1     0     0     4     0
     0     0     2     1    20     0     0     0     0     1
     4     0     0     0     0    21     0     0     0     0
     0     0     0     2     2     0    29     0     1     0
     1     1     2     2     1     1     0    22     1     0
     0     0     1     0     0     0     1     0    30     0
     0     0     0     1     0     0     0     1     0    20
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
```

```
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
```

```
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
```

```
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
clppv =  3
```

Published with GNU Octave 4.4.0