Recognition Algorithmics

Robert B. Fisher

School of Informatics

University of Edinburgh

Matlab code for probability calculation

```
function prob = multivariate(Vec,Mean,...
                            Invcor,apriori)
 diff = Vec-Mean;
 n = length(Vec);
 wgt = sqrt(det(Invcor));
 dist = diff*Invcor*diff';
 prob = apriori * ( 1 / (2*pi)^(n/2)) ...
          * wgt * exp(-0.5*dist);
```

Training

Estimating classifier parameters $p(c), \vec{m}_c, \mathcal{A}_c$

Should split data into:

- Training set - used to estimate parameters (eg. 50% of data)

- Validation set - used to tell when to stop training, or to fix hyper-parameters (eg. 25% of data)

- Test set - used to test performance (eg. 25% of data)

Must have more training samples than property dimension n

Training Code

```
Dim = 3;            % number of feature properties
modelfilename =
    input('Model file name (filename)\n?','s');
maxclasses = input('Number of classes (int)\n?');
trainfilestem = input('Training image file stem
                            (filestem)\n?','s');
N = input('Number of training images (int)\n?');
for imagenum = 1 : N
   currentimagergb = imread([trainfilestem,
            int2str(imagenum),'.jpg'],'jpg');
   currentimage = rgb2gray(currentimagergb);
```

Slide based on: Bob Fisher

```matlab
    vec(imagenum,:) =
            extractprops(currentimage,0,0,0,0,0);
    trueclasses(imagenum) = input(['Train image ',
        int2str(imagenum),' true class (1..',
        int2str(maxclasses),')\n?']);
end
[Means,Invcors,Aprioris] = buildmodel(Dim,vec,N,
                        maxclasses,trueclasses);
eval(['save ',modelfilename,' maxclasses ...
                Means Invcors Aprioris'])
```

Slide based on: Bob Fisher

Building statistical model

```
function [Means,Invcors,Aprioris] = ...
      buildmodel(Dim,Vecs,N,Numclass,Classes)
  for i = 1 : Numclass
    samples = find(Classes == i); % locate cls i
    M = length(samples); % num in class
    classvecs = Vecs(samples,:); % get members
    mn = mean(classvecs);
    Means(i,:) = mn;
    diffs = classvecs - ones(M,1)*mn;
    Invcors(i,:,:) = inv(diffs'*diffs/(M-1));
    Aprioris(i) = M/N;
  end
```

Slide based on: Bob Fisher

Test Code

```
eval(['load ',modelfilename,...
        ' maxclasses Means Invcors Aprioris'])
imagestem = input('Test image file stem ...
                    (filestem)\n?','s');
run=1;
imagenum=0;
while ~(run == 0)
   imagenum = imagenum + 1;
   currentimagergb = imread([imagestem, ...
            int2str(imagenum),'.jpg'],'jpg');
   currentimage = rgb2gray(currentimagergb);
   vec = extractprops(currentimage,0,0,0,0,0);
```

Slide based on: Bob Fisher

```
class=classify(vec,maxclasses,Means,Invcors,...
     Dim,Aprioris)
  run = input(['Want to process another image ',
          int2str(imagenum+1),' (0,1)\n?']);
end
```

Lecture Overview

- Matlab to make ideas concrete

- Code for training, testing and probability calculation

- Good data approach for training, validation and testing

Slide based on: Bob Fisher