Tai Duc Nguyen - ECEC487 - HW1 - 09/29/2019

Contents

Computer Exercises (CX)

```
seed = 0
randn('seed',seed);
```

```
seed =

0
```

HW CX 2.7

```
clear; close all;

% Prepare the variables:
N = 10000;

m1 = [1;1];
m2 = [4;4];
m3 = [8;1];
m = [m1 m2 m3];
```

```
  S1 = eye(2)*2;
  S2 = S1;
  S3 = S1;
  S(:,:,1) = S1; S(:,:,2) = S2; S(:,:,3) = S3;

  P_x5 = [1/3;1/3;1/3];
  P_x5_ = [0.8;0.1;0.1];

  % Generate data set X5 and X5_ and class assignments y5 and y5_
  [X5, y5] = generate_gauss_classes(m, S, P_x5, N);
  [X5_, y5_] = generate_gauss_classes(m, S, P_x5_, N);

  % Classify data set
  z5_bayes = bayes_classifier(m,S,P_x5,X5);
  z5_bayes_ = bayes_classifier(m,S,P_x5_,X5_);
  z5_eu = euclid_classifier(m,X5);
  z5_eu_ = euclid_classifier(m,X5_);

  % Calculate error for the bayes classifiers
  error_x5_bayes = z5_bayes-y5;
  error_x5_bayes(error_x5_bayes > 0) = 1; error_x5_bayes(error_x5_bayes < 0) = 1;
  error_x5_bayes = sum(error_x5_bayes)/length(z5_bayes)

  error_x5_bayes_ = z5_bayes_-y5_;
  error_x5_bayes_(error_x5_bayes_ > 0) = 1; error_x5_bayes_(error_x5_bayes_ < 0) = 1;
  error_x5_bayes_ = sum(error_x5_bayes_)/length(z5_bayes_)

  % Calculate error for the euclidean classifiers
  error_x5_eu = z5_eu-y5;
  error_x5_eu(error_x5_eu > 0) = 1; error_x5_eu(error_x5_eu < 0) = 1;
  error_x5_eu = sum(error_x5_eu)/length(z5_eu)

  error_x5_eu_ = z5_eu_-y5_;
  error_x5_eu_(error_x5_eu_ > 0) = 1; error_x5_eu_(error_x5_eu_ < 0) = 1;
  error_x5_eu_ = sum(error_x5_eu_)/length(z5_eu_)


error_x5_bayes =


  0.0711


error_x5_bayes_ =


  0.0436


error_x5_eu =


  0.0711
```

```
error_x5_eu_ =

  0.0708
```

HW CX 2.8

```
clear; close all;

% Prepare the variables:
N = 1000;

m1 = [1;1];
m2 = [8;6];
m3 = [13;1];
m = [m1 m2 m3];

S1 = eye(2)*6;
S2 = S1;
S3 = S1;
S(:,:,1) = S1; S(:,:,2) = S2; S(:,:,3) = S3;

P = [1/3;1/3;1/3];

% Generate data set X, Z and class assignments y_x, y_z
[X, y_x] = generate_gauss_classes(m, S, P, N);
[Z, y_z] = generate_gauss_classes(m, S, P, N);

% Run KNN classifier with k=1 and k=11
z_knn_1 = k_nn_classifier(Z,y_z,1,X);
z_knn_11 = k_nn_classifier(Z,y_z,11,X);

% Calculate error for the KNN classifiers
error_knn_1 = z_knn_1-y_x;
error_knn_1(error_knn_1 > 0) = 1; error_knn_1(error_knn_1 < 0) = 1;
error_knn_1 = sum(error_knn_1)/length(z_knn_1)

error_knn_11 = z_knn_11-y_x;
error_knn_11(error_knn_11 > 0) = 1; error_knn_11(error_knn_11 < 0) = 1;
error_knn_11 = sum(error_knn_11)/length(z_knn_11)
```

```
error_knn_1 =

  0.1051

error_knn_11 =

  0.0701
```

Using the data generation procedures from textbook CX 2.3 Page 80 with slight modifications

```
function [X,y] = generate_gauss_classes(m,S,P,N)
[~,c]=size(m);
X=[];
y=[];
    for j=1:c
        % Generating the [p(j)*N)] vectors from each distribution
        t=mvnrnd(m(:,j),S(:,:,j),fix(P(j)*N));
        % The total number of points may be slightly less than N
        % due to the fix operator
        X=[X; t];
        y=[y ones(1,fix(P(j)*N))*j];
    end
end
```

Using the Bayes classifier algorithm from textbook CX 2.5 Page 81 with slight modifications

```
function z = bayes_classifier(m,S,P,X)
    [~,c]= size(m); % c=no. of classes
    [N,~]=size(X);  % N=no. of vectors
    t=zeros(c,1);
    z=zeros(1,N);
    for i=1:N
        for j=1:c
            t(j)=P(j)*prob_density_value(X(i,:),m(:,j)',S(:,:,j));
        end
    % Determining the maximum quantity Pi*p(x|wi)
    [~,z(i)]=max(t);
    end
end
```

Using the Euclidean classifier algorithm from textbook CX 2.6 Page 82 with slight modifications

```
function z = euclid_classifier(m,X)
    [~,c]=size(m); % c=no. of classes
    [N,~]=size(X); % N=no. of vectors/
    t=zeros(c,1);
    z=zeros(1,N);
    for i=1:N
        for j=1:c
            t(j)=sqrt((X(i,:)-m(:,j)')*(X(i,:)-m(:,j)')');
```

```
        end
    % Determining the maximum quantity Pi*p(x|wi)
    [~,z(i)]=min(t);
    end
end
```

Calculate the probability density value (using equation from page 30, class's slides ac_1_classifier_bayes_1.ppt)

```
function res = prob_density_value(X,m,S)
    [b,~]=size(m);
    res = 1/((2*pi)^(b/2)*det(S)^(1/2))*exp(-1/2*(X-m)*inv(S)*(X-m)');
end
```

Using the Euclidean classifier algorithm from textbook CX 2.8 Page 82-83 with modifications

```
function z=k_nn_classifier(Z,v,k,X)
    [~,N1]=size(Z);
    [N,~]=size(X);
    c=max(v); % The number of classes
    % Computation of the (squared) Euclidean distance
    % of a point from each reference vector
    z=zeros(1,N);
    for i=1:N
        dist=sum(((repmat(X(i,:),N,1)-Z).^ 2),2);
        %Sorting the above distances in ascending order
        [~,n]=sort(dist);
        % Counting the class occurrences among the k-closest
        % reference vectors Z(:,i)
        refe=zeros(1,c); %Counting the reference vectors per class
        for q=1:k
            class=v(n(q));
            refe(class)=refe(class)+1;
        end
        [~,z(i)]=max(refe);
    end
end
```

Conclusion and remarks

```
error_x5_bayes =

  0.0711
```

```
error_x5_bayes_ =

  0.0436

error_x5_eu =

  0.0711

error_x5_eu_ =

  0.0708

error_knn_1 =

  0.1051

error_knn_11 =

  0.0701
```

From the results of the experiment 1 (CX 2.7), it is observed that Bayes(B) and Euclidean(E) classifier get the same error when the classes' a priori probabilities are equal. However, when the a prior probabilities are heavily skewed to one class, the B classifier performs better. The results of experiment 2 (CX 2.8) show that increasing k from 1 to 11 does increase the performance of the classifier, however, it it not a positively linear relationship between k and performance. The error rate goes even higher than k=1 with k=999 (error=0.6667)

Written Homework

Question

What is a sufficient statistic for the Gaussian, Poisson, and exponential distributions? Express the mean and variance for these distributions in terms of the sufficient statistic

Answer

In the Reading "*On the Mathematical Foundations of Theoretical Statistics*" by R. A. Fisher (1922), page 316-317, the author introduces the **Criterion of Sufficiency**, stating:

> If θ be the parameter to be estimated, θ_1, a statistic which contains the whole of the information as to the value of θ, which the sample supplies, and θ_2 any other statistic, then the surface of distribution of pairs of values θ_1 and θ_2, for a given value of θ, is such that for a given value of θ_1, the distribution of θ_2 does not involve θ.

Hence, we can use *Fisher–Neyman factorization theorem* to check if a statistic is sufficient. The theorem is stating:

> Let $X_1, X_2, \cdots, X_n$ be a random sample with joint density $f(x_1, x_2, \cdots, x_n|\theta)$. A statistic $T = r(X_1, X_2, \cdots, X_n)$ is sufficient if and only if the joint density can be factored as follows:
>
> $f(x_1, x_2, \cdots, x_n|\theta) = u(x_1, x_2, \cdots, x_n)v(r(x_1, x_2, \cdots, x_n), \theta)(2)$
>
> where u and v are non-negative functions. The function u can depend on the full random sample $x_1, \cdots, x_n$, but not on the unknown parameter θ. The function v can depend on θ, but can depend on the random sample only through the value of $r(x_1, \cdots, x_n)$.

For Gaussian distributions*:

If the **mean** is a sufficient statistic, then $f(x_1, x_2, \cdots, x_n|\theta) = f_\theta(x)$ can be described as:

> $\Pi_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x_i-\theta)^2}{2\sigma^2})$

which can be fractorized into

> $f_\theta(x) = (2\pi\sigma^2)^{-n/2} \times exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \bar{x})^2) \times exp(-\frac{n}{2\sigma^2}(\theta - \bar{x})^2)$

Let $u(x) = (2\pi\sigma^2)^{-n/2} exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \bar{x})^2)$ and $v_\theta(x) = exp(-\frac{n}{2\sigma^2}(\theta - \bar{x})^2$ then: $u(x)$ does not depend on θ and $v_\theta(x)$ only depends on $r(x)$ through the function $r(x) = \bar{x}$ -- which is the **mean**

If the **variance** is a sufficient statistic, then $f(x_1, x_2, \cdots, x_n|\theta) = f_\theta(x)$ can be described as:

> $\Pi_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x_i-\theta)^2}{2\sigma^2})$

which can be fractorized into

> $f_\theta(x) = (2\pi\sigma^2)^{-n/2} \times exp(-\frac{\sum_{i=1}^{n}(x_i-\theta)^2}{2\sigma^2})$

Let $u(x) = (2\pi\sigma^2)^{-n/2}$ and $v_\theta(x) = exp(-\frac{\sum_{i=1}^{n}(x_i-\theta)^2}{2\sigma^2})$ then: $u(x)$ does not depend on θ and $v_\theta(x)$ only depends on $r(x)$ through the function $r(x) = \sum_{i=1}^{n}(x_i - \theta)^2$ -- which is the **variance** when take $\frac{r(x)}{n-1}$

For Poisson distributions*:

If the λ is a sufficient statistic, then $f(x_1, x_2, \cdots, x_n|\theta) = f_\theta(x)$ can be described as:

> $e^{-n\times\theta}\theta^{(x_1+...+x_n)} \times \frac{1}{x_1!...x_n!}$

Let $u(x) = \frac{1}{x_1! \dots x_n!}$ and $v_\theta(x) = e^{-n \times \theta} \theta^{(x_1 + \dots + x_n)}$ then: $u(x)$ does not depend on θ and $v_\theta(x)$ only depends on $r(x)$ through the function $r(x) = \sum_{i=1}^{n} x_i$ -- which is the λ when take $\frac{1}{r(x)}$

For Exponential distributions*:

If the λ is a sufficient statistic, then $f(x_1, x_2, \cdots, x_n | \theta) = f_\theta(x)$ can be described as:

> $\Pi_{i=1}^{n} \frac{1}{\theta} exp(-\frac{1}{\theta} x_i)$

which can be fractorized into

> $f_\theta(x) = \theta^{-n} e^{-\frac{1}{\theta} \sum_{i=1}^{n} x_i}$

Let $u(x) = 1$ and $v_\theta(x) = \theta^{-n} e^{-\frac{1}{\theta} \sum_{i=1}^{n} x_i}$ then: $u(x)$ does not depend on θ and $v_\theta(x)$ only depends on $r(x)$ through the function $r(x) = \sum_{i=1}^{n} x_i$ -- which is the λ when take $\frac{1}{r(x)}$

** All the factorizations are done with reference to https://en.wikipedia.org/wiki/Sufficient_statistic*