Breast Cancer Wisconsin (Diagnostic)

1. Abstract

we have features that are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the images
we need to classify the images in two classes to diagnose breast cancer into (Malignant or Benign)

2. Dataset Information:

1. Number of instances: 569

2. Number of attributes: 32 (ID, diagnosis, 30 real-valued input features)

3. Attribute information

   1) ID number
   2) Diagnosis (M = malignant, B = benign)
   3-32) Ten real-valued features are computed for each cell nucleus:

      a) radius (mean of distances from center to points on the perimeter)
      b) texture (standard deviation of gray-scale values)
      c) perimeter
      d) area
      e) smoothness (local variation in radius lengths)
      f) compactness (perimeter^2 / area - 1.0)
      g) concavity (severity of concave portions of the contour)
      h) concave points (number of concave portions of the contour)
      i) symmetry
      j) fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.
All feature values are recoded with four significant digits.

4. Missing attribute values: none

5. Class distribution: 357 benign, 212 malignant

3. Previous Results

First Usage: O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.

Best Result: best predictive accuracy obtained using one separating plane in the 3-D space of Worst Area, Worst Smoothness and Mean Texture. Estimated accuracy 97.5% using repeated 10-fold crossvalidations. Classifier has correctly diagnosed 176 consecutive new patients as of November 1995.

4. Preprocessing on data

replace M with 1 and B with 2, also remove the ID as we didn't need it in our case

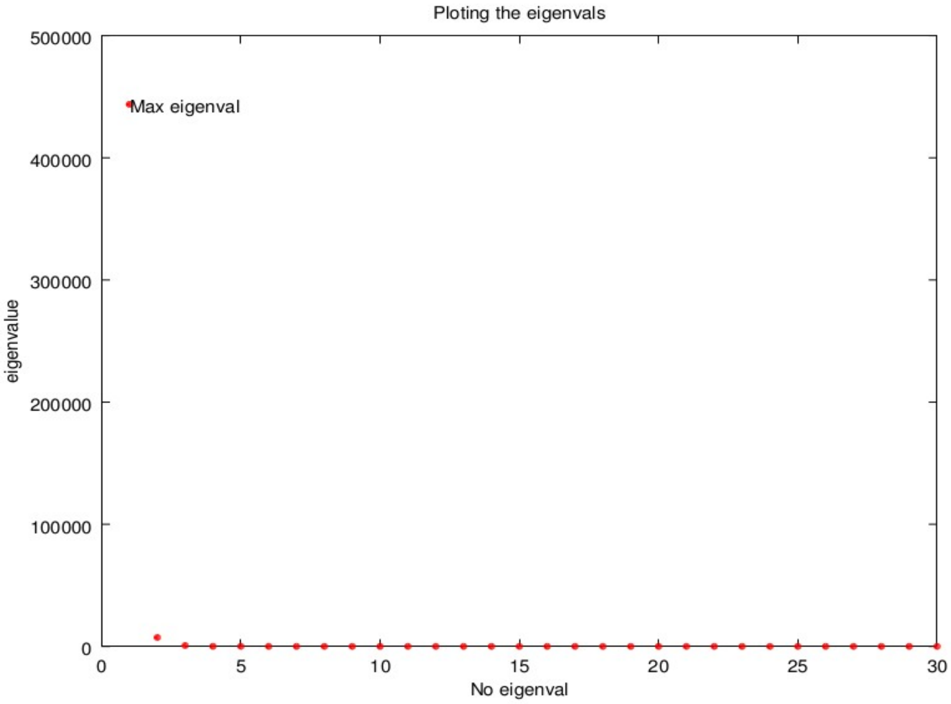
5. Octave code

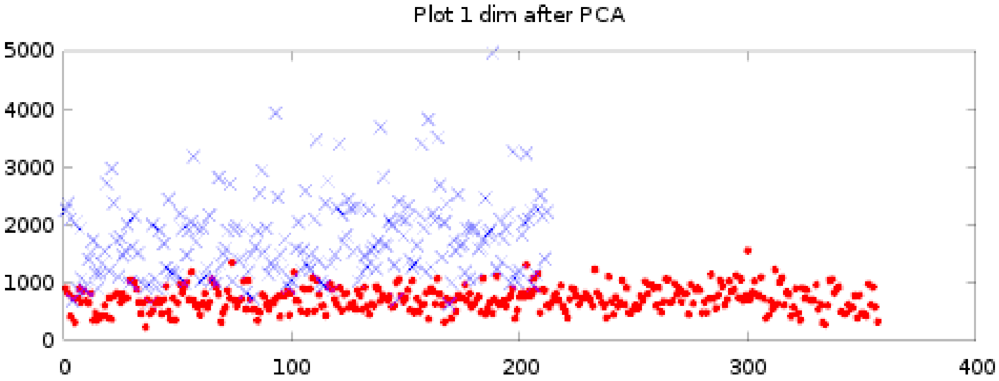
```
%% Initialization
clear ; close all; randn('seed',0);
%% =============== Part 1: Loading Data from test.data ====================
data=load('test.data');
X_data=data(:,3:end);
y_data=data(:,2);
m_features=30;

%% ============ Part 2: Estimating Two Probabilities ===================
N=length(y_data);
m_size=length(find(y_data==1));
b_size=length(find(y_data==2));
P=[m_size b_size]'./N;

%% ============= Part 3: Random Data Split 75% to 25%         ================
rndIdx=randperm(N);
training_range=floor(75*N/100);

X_train=X_data(1:training_range,:)';
y_train=y_data(1:training_range,:)';

X_test=X_data(training_range+1:end,:)';
y_test=y_data(training_range+1:end,:)';

%% =========== Part 4: Estimating Mean and Variance ===================

malignant_data=X_train(:,find(y_train==1));
[m1_hat, S1_hat]=Gaussian_ML_estimate(malignant_data);
benign_data=X_train(:,find(y_train==2));
[m2_hat, S2_hat]=Gaussian_ML_estimate(benign_data);

S_hat=cat(3,S1_hat,S2_hat);
m_hat=[m1_hat m2_hat];

%% =========== Part 5: Applying Bayesian Classifier  ==================
y_bayesian=bayes_classifier(m_hat,S_hat,P,X_test);

%% =========== Part 6: Estimating Error of Bayes  =====================
err_bayesian = (1-length(find(y_test==y_bayesian))/length(y_test));
fprintf('Bayesian classifier y_test error is %.3f%% \n',err_bayesian*100);

%% =============== Part 7: Using PCA  ============================
m=30;
[eigenval,eigenvec,explained,Y,mean_vec]=pca_fun(X_data',m);
figure(1),plot([1:length(eigenval)],eigenval,'.r');
hold on;
title ('Ploting the eigenvals');
```

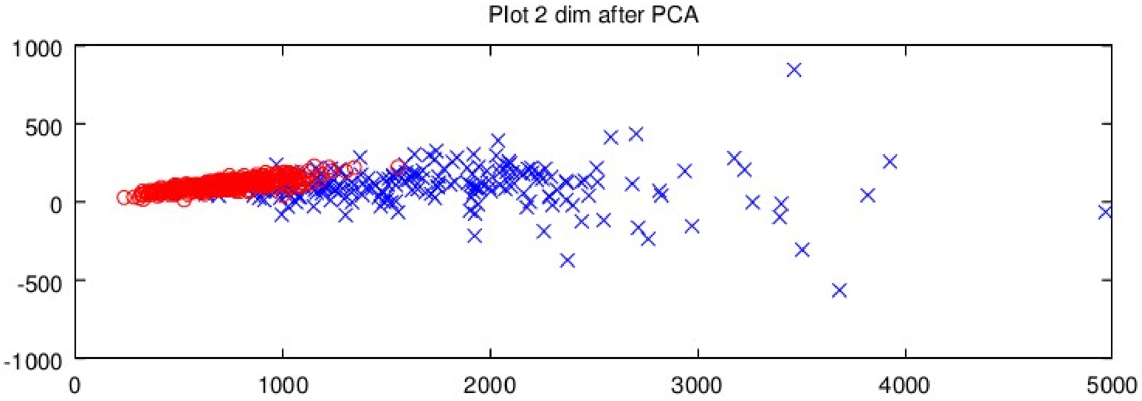
```
text (1,eigenval(1),'Max eigenval');
xlabel ('No eigenval');
ylabel ('eigenvalue');
hold off;

m=2;
[eigenval,eigenvec,explained,Y,mean_vec]=pca_fun(X_data',m);
original_data=y_data';

a=Y(:,original_data==1);
b=Y(:,original_data==2);
figure(2),subplot (2, 1, 1),plot(a(1,:),a(2,:),'xb');
hold on;
title ('Plot 2 dim after PCA');
plot(b(1,:),b(2,:),'or');
hold off;

%% ============= Part 8: Classify using Bayes after PCA ===============

X_data_pca=Y';

X_train=X_data_pca(1:training_range,:)';
y_train=y_data(1:training_range,:)';

X_test=X_data_pca(training_range+1:end,:)';
y_test=y_data(training_range+1:end,:)';

malignant_data=X_train(:,find(y_train==1));
[m1_hat, S1_hat]=Gaussian_ML_estimate(malignant_data);
benign_data=X_train(:,find(y_train==2));
[m2_hat, S2_hat]=Gaussian_ML_estimate(benign_data);

S_hat=cat(3,S1_hat,S2_hat);
m_hat=[m1_hat m2_hat];

y_bayesian=bayes_classifier(m_hat,S_hat,P,X_test);

err_bayesian = (1-length(find(y_test==y_bayesian))/length(y_test));
fprintf('After PCA Bayesian classifier error is %.3f%% \n',err_bayesian*100);

%% ========= Part 9: Using LDA  ========================
X2=X_data';
y2=y_data';
mv_est(:,1)=mean(X2(:,y2==1)')';
mv_est(:,2)=mean(X2(:,y2==2)')';
[Sw,Sb,Sm]=scatter_mat(X2,y2);
w=inv(Sw)*(mv_est(:,1)-mv_est(:,2));

%Computation of the projections
t1=w'*X2(:,y2==1);
t2=w'*X2(:,y2==2);
%Plot of the projections
```

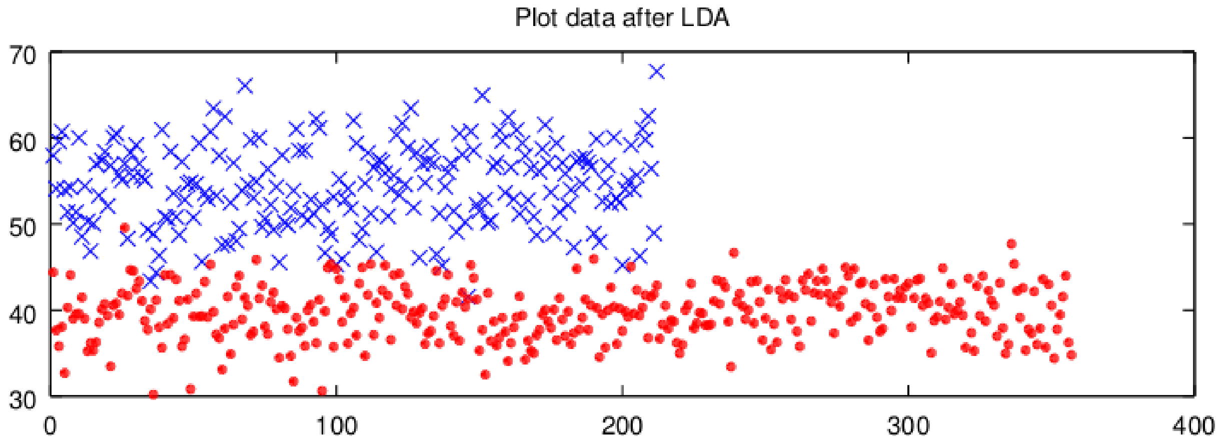
```
subplot (2, 1, 2),plot(t1,'xb');
hold on;
title ('Plot data after LDA');
plot(t2,'.r');
hold off;


%% ============= Part 9: Classify using Bayes after LDA ===============

X_data_lda=w'*X2;
X_data_lda=X_data_lda';

X_train=X_data_lda(1:training_range,:)';
y_train=y_data(1:training_range,:)';

X_test=X_data_lda(training_range+1:end,:)';
y_test=y_data(training_range+1:end,:)';

malignant_data=X_train(:,find(y_train==1));
[m1_hat, S1_hat]=Gaussian_ML_estimate(malignant_data);
benign_data=X_train(:,find(y_train==2));
[m2_hat, S2_hat]=Gaussian_ML_estimate(benign_data);

S_hat=cat(3,S1_hat,S2_hat);
m_hat=[m1_hat m2_hat];

y_bayesian=bayes_classifier(m_hat,S_hat,P,X_test);

err_bayesian = (1-length(find(y_test==y_bayesian))/length(y_test));
fprintf('After LDA Bayesian classifier error is %.3f%% \n',err_bayesian*100);



X=[X_data'; ones(1,N)];
y=y_data';
y(y==2)=-1; % -1 for class two
rho=0.5;
w_ini=zeros(1,m_features+1)';
[w, iter, mis_clas] = perce(X, y, w_ini, rho);

%% ==========================================================
```

6. Results (Figures)









7. Output
on
Bayesian classifier y_test error is 3.497%

After PCA (one dimension) Bayesian classifier error is 6.294%
After PCA (two dimension) Bayesian classifier error is 6.993%
After PCA (four dimension) Bayesian classifier error is 4.196%
After PCA (30 dimension) Bayesian classifier error is 3.497%

After LDA (One Dimension) Bayesian classifier error is 2.098%

8. Conclusion:

1. The first dimension is one who response to change in the data as its eigenvalue is much great with respect to other 29 dimension

2. by increasing now of dimension the error rate converge to be equal to Bayesian classifier without any analysis

3. Best result can obtained after perform linear discriminant analysis is 2.098%