Appendix-A

Data Description

Dataset-I

This Dataset was obtained from Department of Skin & V.D., The H M Patel Centre for Medical Care and Education (Shree Krishna Hospital), Karamsad, Gujarat, India. We have prepared detailed Proforma under the guidance of Head of the Department of Skin and V. D., Dr. Rita Vohra and investigated 470 patients. The proforma includes 47 features. The study includes 47 features and 470 instances. Out of 470 instances 139 instances are for Bacterial Infections, 146 for Fungal Infection, 98 for Eczema and 87 for Scabies. The following table shows various features which are investigated during our data collection. Patients identity is removed from training and testing Dataset.

Attributes Information for Dataset-I

Chief Complaints & OPD

1) Pain 2) Fever 3) Itching

Seasonal relation

4) Summer 5) Winter 6) Monsoon

Past History

7) Diabetes Mellitus 8) Family History

Occupational History

9) Hot and humid environment 10) Exposure to irritants 11) Excessive sun exposure

Type of Lesion

12) Macules 13) Patches 14) Papules

15) Pustule 16) Nodule 17) Plaques

18) Vesicles 19) Bullae

Colour

20) Erythematous 21) Hyperpigmented

22) Hypopigmented

Associated With

23) Lichenification 24) Oozing 25) Crusting

26) Scaling 27)Excoriation 28) Discharge

Shape

29) Linear 30) Annular 31) Grouped

Sites

32) Webspaces 33) Wrist 34) Forearm

Type of Lesion

35) Arm 36) Chest 37) Abdomen

38) Genitals 39) Thigh 40) Legs

41) Dorsa of feet 42) Back 43) Buttocks

44) Palms & Soles 45) Hair 46) Nail 47) Face

Patient Information Sheet & Informed Consent Form

This Information sheet is for women who have recently given birth, and who we are inviting to participate in research. The title of our research project is "An Expert System for Diagnosis of Common Skin Diseases using Soft Computing Techniques"
Name of Principal Investigator:
Name of Organization: Shree Krishna Hospital, Karamsad, Anand, Gujarat - 388325
Introduction:

I am going to give you information and invite you to be part of this research. Before you decide, you can talk to anyone you feel comfortable with about the research. There may be some words that you do not understand. Please ask me to stop as we go through the information and I will take time to explain. If you have questions later, you can ask them of me.
Purpose of the research:
The goal of this research is to see how different infant and maternal factors affect the stress level in mother whose child is admitted in NICU. Collecting this information will help us identify the problems faced by these mothers. These findings can further be used to create intervention programs to address these issues. Participant selection:
We are inviting mothers whose babies are admitted in the NICU to participate in this research. As you are also in such a state so we have approached you for participation in this study.
Voluntary Participation:
Your participation in this research is entirely voluntary. It is your choice whether to participate or not. Whether you choose to participate or not, will not have any effect on the health services you are currently receiving. You may change your mind

later and stop participating even if you agreed earlier.

Procedures and Protocol:

The research will involve only one interview session with you while you are visiting your baby and not involved with any other member of the staff. A trained staff nurse would interview you and will ask you questions about:—— Personal information regarding yourself i.e. name, age, education, income, caste, religion etc. Information regarding your pregnancy, past birth history, emotional experiences during pregnancy and post delivery Data such as the type of delivery, any complications that occurred, baby gender, weight and health status would be collected from your hospital record. Risks:

One of the risks of the study is loss of personal information. This is very unlikely to happen, and we will do everything to make sure that your information is protected.

Benefits:

There may not be any benefit for you but your participation is likely to help us develop interventions that we plan to use in the near future in your community. This may benefit other pregnant women and children in the near future. Confidentiality"

We will not be sharing the identity of those participating in the research. Sharing of information.

The results of the research would be used for academic purpose. We will publish the results in order that other interested people may learn from our research.Your identity will not be revealed in any information released to third parties or published.

Right to Refuse or Withdraw:

You are free to withdraw from this study at any time, without giving any reason, without your medical care or legal rights being affected.

Dataset-II

This data set is from UCI (University of California Irvine) machine learning database[4]. Actual database contains 34 attributes, 33 of which are linear valued and one of them is nominal. We consider only 33 features in our study. The last feature which is the age of patient is omitted.

The diseases in this group are psoriasis, seboreic dermatitis, lichenplanus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. Usually abiopsy is necessary for the diagnosis but unfortunately these diseases share many histopathological features as well. Another difficulty for the differential diagnosis is that a disease may show the features of another disease at the beginning stage and may have some characteristic features in the following stages. Patients were first evaluated clinically with 11 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope. In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases have been observed in the family and 0 otherwise. Every other feature (clinical and histopathological) is given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values. Number of patients in the data set is 366. Patients for Psoriasis is 112, for Seboreic dermatitis is 61, for Lichen planus is 72, for Pityriasis rosea is 49, for Chronic dermatitis is 52 and for Pityriasis rubra pilaris is 20.

Attributes Information for Data Set 2

Clinical Attributes Histopathological Attributes

1) Erythema 2) scaling 3) definite borders

4) itching 5) koebner phenomenon 6) polygonal papules

7) follicular papules 8) Oral mucosal involvement

9) knee and elbow involvement 10) scalp involvement

11) family history, (0 or 1)

Histopathological Attributes

12) melanin incontinence 13) eosinophils in the infiltrate

14) PNL infiltrate 15) fibrosis of the papillary dermis

16) exocytosis 17) Acanthosis 18) hyperkeratosis

19) parakeratosis 20) clubbing of the rete ridges

21) elongation of the rete ridges

22) thinning of the suprapapillary epidermis

23) spongiform pustule 24) munro microabcess

25) focal hypergranulosis 26) disappearance of the granular layer

27) vacuolisation and damage of basal layer 28) spongiosis

29) saw-tooth appearance of retes 30) follicular horn plug

31) perifollicular parakeratosis 32) inflammatory monoluclear inflitrate

33) band-like infiltrate

newff

newff Create a feed-forward backpropagation network

Syntax: net = newff

net = newff(PR,[S1 S2...SNl],TF1 TF2...TFNl,BTF,BLF,PF)

Description:

net = newff creates a new network with a dialog box.

newff(PR,[S1 S2...SNl],TF1 TF2...TFNl,BTF,BLF,PF) takes,

PR - R x 2 matrix of min and max values for R input elements.

Si - Size of i^{th} layer, for Nl layers.

TFi - Transfer function of i^{th} layer, default = 'tansig'.

BTF - Backpropagation network training function, default = 'traingdx'.

BLF - Backpropagation weight/bias learning function, default = 'learngdm'.

PF - Performance function, default = 'mse'.

and returns an N layer feed-forward backprop network.

The transfer functions TFi can be any differentiable transfer function logsig (definition 3.2.2).

The training function BTF can be any of the backprop training functions such as trainlm which use Levenberg-Marquardt backpropagation(algorithm 3.2.2)

HUMAN RESEARCH ETHICS COMMITTEE
H M PATEL CENTRE FOR MEDICAL CARE & EDUCATION
KARAMSAD

HMPCMCE: HREC/2015/8 **Date**: 04/02/2015

To,
Dr. Rita Vora
Professor,
Dept. of Skin & VD,
PSMC, Karamsad

&

Ms. Krupal Parikh
Ph.D Student,
Dept. of Applied Mathematics,
M.S. University, Baroda

Subject: **Approval for your research proposal**

Your proposal entitled "An Expert system for diagnosis of common skin diseases using soft computing techniques" was discussed in the 54th meeting of the Human Research Ethics Committee, H M Patel Centre for Medical Care and Education, Karamsad was held on 15/01/15 between 2:00 pm to 6:30 pm at Academic Block, Pramukhswami Medical College, Karamsad.

Your proposal required modifications based on following suggestions of HREC:

1. Submit fees receipt as per Research group instructions (& approval)
2. Include 'Mentor' for research project
3. References citation to be corrected in objectives
4. Site similar methods, even if done for other conditions
5. Specify that no fund required for project
6. Clarify for what purpose the technique shall be used
7. Clarify on this being a multi-centric study; Point. No. 8 in proposal
8. Request for waiver of consent
9. Specify that data shall be provided without identification parameters

In response to your compliance satisfying the queries raised, HREC approves the research work to be conducted in its prescribed format with the following conditions: HREC expects to be informed about (A) Any change in the protocol (B) Any other ethical issue in course of the research study (C) Report of the study at the end

Please quote the reference number "**HMPCMCE: HREC/2015/8"** in all future communications.

Swapnil 04/02/15
Sign with Date

Name: Dr. Swapnil Agarwal
Member Secretary
Human Research Ethics committee
H M Patel Centre for Medical Care and Education,
Karamsad – 388 325
Gujarat, India
Stamp of the HREC

H M Patel Centre for Medical Care & Education
HREC
Karamsad

PROFORMA

A study of skin diseases: SCABIES/ECZEMA/BACTERIAL & FUNGAL INFECTIONS

Department of Skin & VD, Shrikrishna Hospital, Karamsad,Gujarat

Ms. Krupal S.Parikh Dr. Rita Vora

Registration No: Date:_____

BIO DATA:

1. Name: 5. Religion:
2. Age/sex: 6. Marital Status:
3. Education: 7. Income:
4. Occupation:

8. Socioeconomic status

Overcrowding

Malnutrition

9. CHIEF COMPLAINTS & ODP:

Duration

I. Pain II. Fever

III. Itching:

Diurnal variation:

increased at day night

IV. Seasonal relation:

Summer Winter Monsoon

V. Skin lesions

Sites: B/L UL ☐ Back & buttocks ☐
B/L LL ☐ Genitals ☐
Axilla ☐ Palms & soles ☐

Chest & Abdomen ☐

10. Treatment history**:**

Over the counter drugs ☐ Self treatment ☐
Prescribed by doctor ☐

11. Past history**:**

History of Scabies ☐ Eczema ☐

Bacterial infection ☐ Fungal infection ☐
Diabetes Mellitus ☐ Asthma ☐
Hypertension ☐ Tuberculosis ☐
History of atopy ☐

12. Family history:

History of Scabies ☐ Eczema ☐

Bacterial infection ☐ Fungal infection ☐
Diabetes Mellitus ☐ Asthma ☐
Hypertension ☐ Tuberculosis ☐
History of atopy ☐

13. Occupational history:

Hot and humid environment ☐

Exposure to irritant ☐

Excessive sun exposure ☐

14. Personal history:

Addictions:

Smoking ☐ Alcohol ☐ Tobacco Chew ☐

15. Menstrual history:

16. Obstetric history:

CUTANEOUS EXAMINATION:

17. Type of lesion:

Macules ☐ Patches ☐ Papules ☐

Nodule ☐ Plaques ☐ Vesicles ☐

Bullae ☐ Pustule ☐

18. Colour:

Erythematous ☐ Hyperpigmented ☐

Hypopigmented ☐

19. Associated with:

Lichenification ☐ Oozing ☐ Crusting ☐

Scaling ☐ Excoriation ☐ Discharge ☐

20. Shape:

Linear ☐ Annular ☐

Grouped ☐ Reticular ☐

21. Size of lesion:

22. Sites:

UL: Webspaces ☐ LL: Thigh ☐

Wrist ☐ Legs ☐

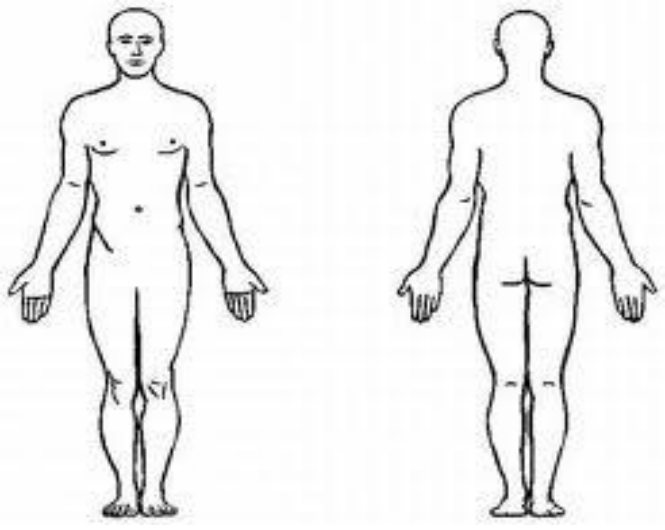
Forearm ☐ Dorsa of feet ☐

Arm ☐

Chest ☐ Back ☐

Abdomen ☐ Buttocks ☐

Genitals ☐ Palms & Soles ☐



23. Hair:

24. Nail:

25. Oral Mucosa:

26. CLINICAL DIAGNOSIS :

Bacterial infection	☐
Fungal infection	☐
Eczema	☐
Scabies	☐
Others	☐

દર્દી માહિતી પત્રક

આ માહિતી પત્રક ખૂજલી/ખરજવું/બેકટેરિયા અને ફંગલ ચેપ પીડાતા ત્વચા દર્દીઓ માટે છે, અને હું આ સંશોધન અભ્યાસમાં ભાગ લેવા માટે આમંત્રિત કરું છું. મારા સંશોધન પ્રોજેકટ શીર્ષક ''સોફ્ટ કમપ્યુટિંગ તકનીકો ઉપયોગ કરીને સામાન્ય સ્કિન ડિસીઝ નિદાન માટે એક એકસપર્ટ સિસ્ટમ'' છે.

મુખ્ય તપાસનીય નામ :

સંસ્થાનું નામ : શ્રી કૃષ્ણા હોસ્પીટલ, કામસદ, ગુજરાત

પરિચય :

હું તમને મારા સંશોધન વિશે જણાવું છું અને તમે મારા સંશોધનનો ભાગ બનો તે માટે તમને આમંત્રિત કરું છું. આમાનાં ઘણા શબ્દો તમે સમજી નહી શકો. જો માહિતી દરમ્યાન તમને કોઈપણ બાબતે સમજ ના પડે તો મને રોકજો હું તમને સમય લઈ સમજાવીશ તમને વધારે પ્રશ્નો હોય તો પાછળથી પણ મને પૂછી શકો છો.

સંશોધનનો હેતુ :

ત્વચા રોગો વિશ્વમાં હવે ખૂબ જ સામાન્ય છે. વિવિધ પ્રકારની એલર્જી પણ વધુ સામાન્ય બની રહી છે. ત્વચાનાં રોગો સામાન્ય રીતે પ્રાથમિક આરોગ્ય કેન્દ્રો, સામુદાયિક સ્વાસ્થ્ય કેન્દ્રો, રેફરલ હોસ્પિટલ ખાતે મેડિકલ અને પેરામેડિકલ સ્ટાફ તેમજ વિશિષ્ટ હોસ્પિટલો દ્વારા કરવામાં આવે છે. આ રોગો ઘણા પ્રારંભિક તબક્કે સારવાર ન કરવામાં આવે તો ખૂબ જ ખતરનાક છે. અયોગ્ય નિદાનને કારણેઘણી વખત સારવાર ખોટી રીતે અને સ્થાનિક એન્ટીબેકટેરિયલ, ફૂગપ્રતિરોધી અને સ્ટીરોઈડ મિશ્રણ દ્વારા કરવામાં આવે છે. આ સારવાર સમાજમાં જોખમી છે અને સ્ટિરોઈડ જેવી સ્થાનિક એજન્ટોની આડઅસરો આવે છે. આ વિકૃતિઓ માટે યોગ્ય નિદાન ચોક્કસ સારવાર માટે જરૂર છે. આ સંશોધન દ્વારા અમે ચામડીના રોગો નિદાન કરી શકે એ નિષ્ણાત સિસ્ટમ વિકસાવવા પ્રયાસ કરી રહ્યા છે.

સ્વૈચ્છિક ભાગીદારી :

આ સંશોધનમાં તમારી ભાગીદારી સંપૂર્ણપણે સ્વૈચ્છિક છે.

પ્રક્રિયા અને પદ્ધતિ :

આ સંશોધન તમારી સાથે માત્ર એક મુલાકાતમાં થશે. આ તપાસનીસ તમને પ્રશ્નો પૂછશે :

- વ્યકિતગત માહિતી, તમારી જાતને સંબંધિત એટલે કે નામ, ઉંમર, શિક્ષણ વગેરે

– તમારી મુખ્ય ફરિયાદ, સારવાર ઈતિહાસ અંગેની માહિતી, ભૂતકાળ ઈતિહાસ, રોગ અંગે કુટુંબ ઈતિહાસ વગેરે

ફાયદાઓ :

આ સંશોધનમાં ભાગ લેવાથી તમને ફાયદો થવાની શકયતા ઓછી છે. પણ તમારી ભાગીદારી અમને ત્વચા રોગ નિદાન નિષ્ણાત સિસ્ટમ વિકસાવવા મદદ કરી શકે છે. નિષ્ણાત ડોકટરો જ્યાં નથી ત્યાં અને ગ્રામીણ વિસ્તારમાં રહેતા લોકોને લાભ થઈ શકે છે.

ગુપ્તતા :

તમે સંશોધનનો ભાગ છે જે ઓળખ પ્રકાશિત કરવામાં આવશે નહીં.

સંશોધન માંથી મેળવેલી માહિતી વિતરણ :

આ સંશોધનના પરિણામ શૈક્ષણિક હેતુ માટે ઉપયોગ કરવામાં આવશે અને તેનાં તારણો અન્ય રસ ધરાવતા લોકો માટે પ્રકાશિત કરવામાં આવશે પણ તમારી ઓળખ જાહેર કરવામાં આવશે નહીં.

બાકાત રાખવાનો હક :

તમે કોઈ પણ કારણ આપ્યા વગર સંશોધન માં થી પોતાને બાકાત રાખી શકો છો આ કરવાથી તમારી સારવાર અથવા કાનૂની અધિકારો પર કોઈ અસર પડશે નહીં.

સમંતિપત્ર

આથીહુંસહીકરનાર _____ સમંતિઆપુછુંકે"સોફ્ટ કમ્પ્યુટિંગ તકનીકો ઉપયોગ કરી

સામાન્ય સ્કિન ડિસીઝ નિદાન માટે એક એક્સપર્ટ સિસ્ટમ"સંશોધન વિષયક અભ્યાસમા મારેશો ભાગ ભજવ

છે. તેની ડોક્ટર દ્વારા મને મારી ભાષામાં બરોબર માહિતી આપેલ છે. મારા તમામ પ્રશ્નોનો સંતોષકારક રી

આપવામાંઆવેલછે.

હુ મારી બિમારી વિશે બધી જરુરી માહીતી આપવા તથા સંશોધન માટે જરુરી ફોટા પાડવા સંમતી આપુ છુ

મનેપણસમજણઆપેલછેકે,મારાદ્વારાઆપવામાંઆવતીમાહિતીઅત્યંતગુ ંતરાખવામાંઆવશેઅનેમનેઆજસુધ

લકેહવેપછીમળવાપાત્રસેવાઓપરતેનીકોઈઅસરથશેનહી.

હુંકોઈપણસમયેઆસંશોધનછોડવામુક્તંછું.

નામ_____ઉમર_____વ ંપુરુષ/સ્ત્રી

કમાક_____

સહી(દર્દીઅથવાસાક્ષીની)_____

ડાબાઅંગુઠાનુનીશાન

તા:_____ સ્થળ_____

H M Patel Centre for Medical Care & Education

29/10/2015

TO WHOM SO EVER IT MAY CONCERNS.

It is certify that Krupal Parikh had collected data for "An Expert-System at diagnosis at common Skin diseases using soft Computing techique" for her research Under my guidance & supervision from the dept of Skin. V.D. Shree Krishna Hospital; Karamsad, Gujarat.

Thanks.

R.V.Vora

DR. RITA VIPUL VORA
M.D. (SKIN & V.D.)
PROFESSOR & HEAD,
DEPT. OF SKIN & V.D.
P.D.U. GOVT. MEDICAL COLLEGE
KARAMSAD-388 325, GUJARAT

Pramukhswami Medical College • H M Patel Institute of Post Graduate Studies • G H Patel School of Nursing
K M Patel Institute of Physiotherapy • Smt. L P Patel Institute of Medical Laboratory Technology
Hospital & Medical Research Centre • Manibhai Shivabhai Patel Cancer Centre • Bhanubhai & Madhuben Patel Cardiac Centre
rogya Mandal, Gokal Nagar, Karamsad - 388 325, Gujarat (India) • Phone: 02692 - 222130 & 222587 • Fax: 02692 - 223466 • www.charutarhealth.org

50058

Appendix-B

MATLAB Programs

1. Program to diagnose skin diseases of the Dataset-I using ANN:

```
%Program to classify the data using ANN with 2 hidden layers
clc;
clear all;
f=0;
acc=0;
for i=1:50
fprintf('read input values for 70-30% data set\n');
P_tr=xlsread('H:\Masterdata\P_spliting_70-30.xlsx');
fprintf('read output values for corresponding inputs');
T_tr=xlsread('H:\Masterdata\T_spliting_70-30.xlsx');
%============================================================
fprintf('read input values for 80-20% data set\n');
P_tr=xlsread('H:\Masterdata\P_spliting_80-20.xlsx');
fprintf('read target values for correspoing inputs');
T_tr=xlsread('H:\Masterdata\T_spliting_80-20.xlsx');
```

```
%============================================================
P=P_tr';
T=T_tr';
% % normalize to -1...1
 P = ( (-minP)./(maxP-minP) - 0.5 ) *2
 T = ( (-minT)./(maxT-minT) - 0.5 ) *2

%network for two hidden layers
 net=newff(P,T,[20,10],{'logsig','logsig'});

 net.trainParam.epochs=1000000;
 net.trainParam.goal=0.0000001;
 net.divideFcn = 'divideind';
 net.divideParam.trainInd = 1:329;
 net.divideParam.testInd = 330:470;
%following lines for random division
  net.divideparam.trainratio=0.70;
  net.divideparam.testratio=0.30;
  net.divideparam.valratio=0.0;
 [net,tr]= train(net,P,T);
 outputs=net(P);
performance=perform(net,T,outputs);
 y=sim(net,P);
 plotconfusion(T,y)
[c,cm,ind,per]=confusion(T,y);
fprintf('percentage correct classification :%f \n',100*(1-c));
%  fprintf('percentage incorrect classification:%f \n',100*c);
```

```
for i=1:4
    TP(i)=cm(i,i);
    FP(i)=sum(cm(:,i))-cm(i,i);
    FN(i)=sum(cm(i,:))-cm(i,i);
    T=sum(cm(:,:));
    sumr=0;
    for j=1:4
        sumr=sumr+T(j);
    end
    TN(i)=sumr-sum(cm(:,i))-sum(cm(i,:));
end
sumtp=0;
sumfp=0;
sumfn=0;
sumtn=0;
for i=1:4
    sumtp=sumtp+TP(i);
    sumfp=sumfp+FP(i);
    sumfn=sumfn+FN(i);
    sumtn=sumtn+TN(i);
end
precision=(sumtp/(sumtp+sumfp));
recall=(sumtp/(sumtp+sumfn));
f=2*(precision*recall)/(precision+recall);
acc=(sumtp+sumtn)/(sumtn+sumtp+sumfp+sumfn);
Fscore=f+Fscore;
```

```
 Acc_confusion = acc + Acc_congusion;
end
F_Score = Fscore/50;
Acc_confusion = Acc_confusion/50;
```

2. Program to diagnose skin diseases of the Dataset - I using SVM (with various Positive Definite Kernels)

```
%Program to find accuracy, F_score and G-score using various
%positive definite/conditional positive definite kernel
%functions in SVM
clc;
s=xlsread('H:\Masterdata\Training_70.xlsx');
train_labels = s(:,1);
train_features = s(:,2:end);
%=====================================================
%following lines are to find parameter values
%using grid search method
%=====================================================
features_sparse = sparse(train_features);
 bestcv = 0;
for log2c = -5:15,
  for log2g = -15:0.1:15,
    for d = 1,5
 cmd = ['-v 10 -c ', num2str(2^log2c), ' -g ',
 num2str(2^log2g),'-d',num2str(d)];
 cv = svmtrain(train_labels, train_features, cmd);
  if (cv >= bestcv)
```

```
    bestcv = cv;
    bestc = 2^log2c;
    bestg = 2^log2g;
    bestd=d;
  end
 end
 end
end
fprintf('bestc=%g   bestg=%g bestd=%g', bestc,bestg,bestd)
%============================================================
 st = xlsread('H:\Masterdata\Testing_30.xlsx');
 test_features = st(:,2:end);
 test_labels = st(:,1);
 numTrain = size(train_features,1);
 numTest = size(test_features,1);
 sigma=0.1;
 d1=3;
 d2=2;
 alpha =2;
 c=1;
%radial basis function: exp(-gamma*|u-v|^2)
rbfKernel = @(X,Y) exp(-sigma .* pdist2(X,Y,'euclidean').^2);
%=================================================================
%polyKernel=@(X,Y) (alpha*X*Y'+c).^d1;
%linKernel=@(X,Y)(X*Y'+c);
%t_student_Ker=@(X,Y) (1./(1.+(pdist2(X,Y,'euclidean')).^d2))
%inv_multi_Ker=@(X,Y) 1./(sqrt(((pdist2(X,Y,'euclidean')).^2)+c.^2))
```

```
%compute kernel matrices between every pairs of (train,train) and
%(test,train) instances & include sample serial number as Ist column
%=====================================================================
 K =[(1:numTrain)',rbfKernel(train_features,train_features)];
 KK =[(1:numTest)',rbfKernel(test_features,train_features)];
%=====================================================================
% K =[(1:numTrain)', inv_multi_Ker(train_features,train_features)];
% KK =[(1:numTest)', inv_multi_Ker(test_features,train_features)];
% K=[(1:numTrain)',linKernel(train_features,train_features)];
% KK =[(1:numTest)'  , linKernel(test_features,train_features)];
% K = [(1:numTrain)' , t_student_Ke(train_features,train_features)];
% KK =[(1:numTest)'  , t_student_Ker(test_features,train_features)];
% K = [(1:numTrain)' , polyKernel(train_features,train_features)];
% KK = [(1:numTest)'  , polyKernel(test_features,train_features)];
%=====================================================================
   % % % % %# train and test c=100
 model_ksp = svmtrain(train_labels, K, '-t 4 -c 100');
 [predClass, acc, decVals] = svmpredict(test_labels, KK, model_ksp);
%=====================================================================
 C=confusionmat(test_labels,predClass)
     for i = 1:4
       TP(i) = C(i,i);
       FP(i) = sum(C,2)-C(i,i)
       FN(i) = sum(C(i,:),2)-C(i,i);
       T = sum(C(:,:));
       sumr = 0;
```

```
        for j = 1:4
           sumr = sumr+T(j);
        end
        TN(i) = sumr+C(i,i)-sum(C(:,i))-sum(C(i,:));
        fprintf('\n tp(%d)=%d',i,TP(i));
        fprintf('\n tn(%d)=%d',i,TN(i));
        fprintf('\n fp(%d)=%d',i,FP(i));
        fprintf('\n fn(%d)=%d',i,FN(i));
     end
    sumtp = 0;
    sumfp = 0;
    sumfn = 0;
    sumtn = 0;
for i=1:4
  sumtp = sumtp+ TP(i);
  sumfp = sumfp+FP(i);
  sumfn = sumfn+FN(i);
  sumtn = sumtn+TN(i);
end
precision=(sumtp/(sumtp+sumfp));
recall=(sumtp/(sumtp+sumfn));%sensitivity
spec=(sumtn/(sumtn+sumfp));  %specitivity
Acc_confusion = (sumtn+sumtp)/(sumtn+sumtp+sumfp+sumfn)
f_score = 2*(precision*recall)/(precision+recall)
g_score = sqrt(recall*spec)
```

3. Program to diagnose skin diseases of the Dataset - I using SVM (with various distance based Kernels)

```
%  Program to find classification accuracy(F-Score) using various
%  kernels(PD/indefinite).
clc;
clear all;
s=xlsread('H:\Masterdata\Training_70.xlsx');
train_labels=s(:,1);
train_features=s(:,2:end);
st=xlsread('H:\Masterdata\Testing_30.xlsx');
test_features=st(:,2:end);
test_labels=st(:,1);
numTrain = size(train_features,1);
numTest= size(test_features,1);
sigma=1;
c=1;
alpha=0.2;
%for difference distances
%===============City Block:
 kernel_train=cityblock(train_features,train_features);
 kernel_test=cityblock(test_features,train_features);
 e1=eig(kernel_train);
 pause
 %============Euclidean L2
 rbfKernel= @(X,Y)  pdist2(X,Y,'euclidean');
 kernel_train=(pdist2(train_features,train_features)).^2;
```

```
 kernel_test=(pdist2(test_features,train_features)).^2;
 e1=eig(kernel_train)
 pause
 %=============Minkowski
 %kernel_train= Minkowski(train_features,train_features);
% kernel_test= Minkowski(test_features,train_features);
% e=eig(kernel_train)
 %============== L1family
 %kernel_train= L1family(train_features,train_features);
% kernel_test= L1family(test_features,train_features);
% e=eig(kernel_train)
%==============Intersection
 %kernel_train= intersection(train_features,train_features);
% kernel_test= intersection(test_features,train_features);
% e=eig(kernel_train)
 %============Chi-squares family
 %kernel_train= chisquare_family(train_features,train_features);
% kernel_test= chisquare_family(test_features,train_features);
% e=eig(kernel_train)
 %============Inner Product family
%kernel_train= innerproduct(train_features,train_features);
% kernel_test= innerproduct(test_features,train_features);
% e=eig(kernel_train)
 %===============modiGaussian
%kernel_train= modiGaussian(train_features,train_features);
%kernel_test= modiGaussian(test_features,train_features);
%e=eig(kernel_train)
```

```
%===============Training and Testing
K = [(1:numTrain)' , kernel_train];
KK = [(1:numTest)'  ,kernel_test];
model = svmtrain(train_labels, K, '-t 4 -c 128 -h 0');
[predClass, acc, decVals] = svmpredict(test_labels, KK, model);
```

- Function file to find kernel from cityblock distance

```
function [kernel] = cityblock(a,b)
kernel=zeros(size(a,1),size(b,1));
bsize=size(b,1);
for i=1:bsize
  d=bsxfun(@minus,a,b(i,:));
  kernel(:,i)=(sum(abs(d),2));
end
end
```

- Function file to find various distances from Minkowski Family

```
function [kernel] = Minkowski(a,b)
kernel=zeros(size(a,1),size(b,1));
bsize=size(b,1);
for i=1:bsize
 d=bsxfun(@minus,a,b(i,:));
 kernel(:,i)=(sum(abs((d.^2)),3)).^(1/3);
end
end
```

- Function file to find various distances from L_1 Family

```
function [kernel] = L1family(a,b)
kernel=zeros(size(a,1),size(b,1));
bsize=size(b,1);
for i=1:bsize
 dminus=bsxfun(@minus,a,b(i,:));
 dplus=bsxfun(@plus,a,b(i,:));
 dmax=bsxfun(@max,a,b(i,:));
 dmin=bsxfun(@min,a,b(i,:));
 %sorensen
kernel(:,i)=(sum(abs(dminus),2))./(sum(abs(dplus),2));
%Gower
%kernel(:,i)=(sum(abs(dminus),2))./(size(b,2));
%soergel
%kernel(:,i)= (sum(abs(dminus),2))./(sum(abs(dmax),2));
%Kulczynski
%kernel(:,i)=(sum(abs(dminus),2))./((sum(abs(dmin),2))+0.001);
%Canberra
%kernel(:,i)=sum((abs(dminus)./(dplus+0.001)),2);
%Lorentzian
%kernel(:,i)=sum(log(1+abs(dminus)),2);
end
```

- Function file to find kernels from various distances from Intersection Family

```
function [kernel] = intersection(a,b)
kernel=zeros(size(a,1),size(b,1));
bsize=size(b,1);
```

```
for i=1:bsize
    dminus=bsxfun(@minus,a,b(i,:));
    dplus=bsxfun(@plus,a,b(i,:));
    dmax=bsxfun(@max,a,b(i,:));
    dmin=bsxfun(@min,a,b(i,:));
 %Wave Hedges
 kernel(:,i)=(sum(abs(dminus),2))./(sum((dmax),2));
 %Intersection
 %kernel(:,i)=(0.5)*(sum(abs(dminus),2));
 %Tanimoto
 %kernel(:,i)=sum((dmax-dmin),2)./sum(dmax,2);;
end
```

- Function file to find kernels for various distances from Chisquare Family

```
function [kernel] = chisquare_family(a,b)
%Squared Euclidean
for j=1:bsize
 d = bsxfun(@minus, a, b(j,:));
 kernel(:,j)=sum(d.^2,2);
end
%squared chisquared
%for j=1:bsize
%d = bsxfun(@minus, a, b(j,:));
% s = bsxfun(@plus, a, b(j,:));
%kernel(:,j)=(2.*sum((d.^2./s),2))+0.001;
%end
end
```

- Function file to find kernel for various distances from Inner Product Family

```
function [kernel] = innerproduct(a,b)
kernel=zeros(size(a,1),size(b,1));
k1=zeros(size(a,1),size(b,1));
k2=zeros(size(a,1),size(b,1));
bsize=size(b,1);
asize=size(a,1);
%Standard Inner Product
ddot=a*b';
kernel=ddot;
%cosine
for i=1:asize
for j=1:bsize
kernel(i,j)=(ddot(i,j))./sqrt((asquare(i)).*sqrt(bsquare(j)));
%PCE
%kernel(i,j)=(ddot(i,j))./(asquare(i)+bsquare(j)-ddot(i,j));
end
end
end
```

- Function file to for Modified Gaussian Kernel

```
function[kernel] = modiGaussian(a,b)
kernel=zeros(size(a,1),size(b,1));
bsize=size(b,1);
asize=size(a,1);
%=======================================
```

```
 for i=1:bsize
  for j=1:asize
    d(j,:)=(a(j,:)-b(i,:)).^2 ;
  end
  k(1:asize,i)=sum(d,2);
end
k2=k;
for i=1:asize
  for j=1:bsize
    k(i,j)=k(i,j)+0.001;
end
end
sigma=1.0;
for i=1:asize
  for j=1:bsize
  k1(i,j)=sqrt(k(i,j));
  k2(i,j)=exp((-sigma)*k2(i,j));
  kernel(i,j)=k2(i,j).*(k1(i,j));
 end
end
end
```

- Program to calculate weighted probability for data set-I

```
 %Program to calculate weighted probability for dataset-I
clc;
clear all;
sb=xlsread('H:\Masterdata\mix\Bacterial.xlsx');
```

```
sf=xlsread('H:\Masterdata\mix\Fungal Infection.xlsx');
se=xlsread('H:\Masterdata\mix\Eczema.xlsx');
ss=xlsread('H:\Masterdata\mix\Scabies.xlsx');
w1=470/139;
w2=470/146;
w3=470/98;
w4=470/87;
sumw=w1+w2+w3+w4;
w1=w1/sumw;
w2=w2/sumw;
w3=w3/sumw;
w4=w4/sumw;
for i=1:47
    pb(i)=0;
    pf(i)=0;
    pe(i)=0;
    ps(i)=0;
end
for i=1:47
    for j=1:139
        if(sb(j,i)==1)
            pb(i)=pb(i)+1;
        end
    end
    for j=1:146
        if (sf(j,i)==1)
            pf(i)=pf(i)+1;
```

```
            end
        end
        for j=1:98
            if(se(j,i)==1)
                pe(i)=pe(i)+1;
            end
        end
        for j=1:87
            if(ss(j,i)==1)
                ps(i)=ps(i)+1;
            end
        end
            end

for i=1:47
    prob_b(i)=pb(i)/139;
    prob_f(i)=pf(i)/146;
    prob_e(i)=pe(i)/98;
    prob_s(i)=ps(i)/87;
end

for i=1:47
wp(i)=prob_b(i)*w1+prob_f(i)*w2+prob_e(i)*w3+prob_s(i)*w4;
end
wpsum=0;
for i=1:47
 wpsum=wpsum+wp(i);
```

```
end
wpavg=wpsum/47;
```

- Program to find F_score for data set-I.

```
%Program to calculate F-scores for data set-I
clc;
clear all;
%Input features of whole dataset as well as
% features of individual feature
sdata=xlsread('H:\whole_datase_for_Fscore.xlsx');
sb=xlsread('H:\mix\Bacterial.xlsx');
sf=xlsread('H:\mix\Fungal Infection.xlsx');
se=xlsread('H:\mix\Eczema.xlsx');
ss=xlsread('H:\mix\Scabies.xlsx');
for i=1:47
  pb(i)=0;
  pf(i)=0;
  pe(i)=0;
  ps(i)=0;
end
for i=1:47
  whole(i)=0;
end
for i=1:47
   for j=1:470
     if(sdata(j,i)==1)
      whole(i)=whole(i)+1;
```

```
    end
end
end
for i=1:47
   whole(i)=whole(i)/470;
end
%sum of number of positives for each feature for each disease
for i=1:47
  for j=1:139
    if(sb(j,i)==1)
     pb(i)=pb(i)+1;
    end
end
  for j=1:146
    if (sf(j,i)==1)
     pf(i)=pf(i)+1;
  end
 end
   for j=1:98
     if(se(j,i)==1)
       pe(i)=pe(i)+1;
     end
  end
 for j=1:87
   if(ss(j,i)==1)
     ps(i)=ps(i)+1;
   end
```

```
 end
 end
%=avg of positive for each feature for each disease.
 for i=1:47
  prob_b(i)=pb(i)/139;
  prob_f(i)=pf(i)/146;
  prob_e(i)=pe(i)/98;
  prob_s(i)=ps(i)/87;
 end
%===========================
%difference of avg. of whole data set from average of
%individual disease for each feature.
for i = 1:47
  bavg(i)=(whole(i)-prob_b(i)).^2;
  favg(i)=(whole(i)-prob_f(i)).^2;
  eavg(i)=(whole(i)-prob_e(i)).^2;
  savg(i)=(whole(i)-prob_s(i)).^2;
end
%==================
%================numerator of F-score.
for i=1:47
 num(i)=bavg(i)+favg(i)+eavg(i)+savg(i);
end
%================================
%difference of each feature of a particular disease
%and average of positive feaures of each disease.
for i=1:47
```

```
  bsum(i)=0;
  fsum(i)=0;
  esum(i)=0;
  ssum(i)=0;
end
for i=1:47
  for j=1:139
    bdeno(j,i)=(sb(j,i)-prob_b(i))^2;
    bsum(i)=bsum(i)+bdeno(j,i);
  end
  bsum(i)=bsum(i)/138;
  for j=1:146
   fdeno(j,i)=(sf(j,i)-prob_f(i))^2;
   fsum(i)=fsum(i)+fdeno(j,i);
  end
 fsum(i)=fsum(i)/145;
  for j=1:98
    edeno(j,i)=(se(j,i)-prob_e(i))^2;
    esum(i)=esum(i)+edeno(j,i);
  end
 esum(i)=esum(i)/97;
 for j=1:87
  sdeno(j,i)=(ss(j,i)-prob_s(i))^2;
  ssum(i)=ssum(i)+sdeno(j,i);
 end
  ssum(i)=ssum(i)/96;
end
```

```
for i=1:47
 deno(i)=bsum(i)+fsum(i)+esum(i)+ssum(i);
end
 for i=1:47
   F(i)=num(i)./deno(i);
 end
```

4. Program to find accuracy of sorted data set of weighted probability approach.

```
 clc;
%program to find 10 fold cv and using the parameter value find
%accuracy of the sorted  data set according to
%weighted probability approach.
s=xlsread('H:\whole_data_ksp_wipi');
%parameter values using 10 folds cross validation for 19 sorted
%features according to weighted probability approach.
%For other data use similar algorithm with different data set.
train_labels=s(1:329,1);
train_features=s(1:329,2:end);
test_labels=s(330:end,1);
test_features=s(330:end,1);
%features_sparse=sparse(features);
% bestcv = 0;
%for log2c = -5:15,
%  for log2g = -15:0.1:3,
%cmd = ['-v 10 -c ', num2str(2^log2c), ' -g ', num2str(2^log2g)];
% cv = svmtrain(labels, features, cmd);
%   if (cv >= bestcv)
```

```
     bestcv = cv; bestc = 2^log2c; bestg = 2^log2g;
    end
%  end
%  end
% fprintf('bestc=%g   bestg=%g ',bestc,bestg)

ksp_model=svmtrain(train_labels,train_features,
'-s 0 -t 2 -c 128  -g 0.812252 ');
[predict_labels,accuracy,dec_value]
=svmpredict(test_labels,train_features,ksp_model);
```

5. Program to diagnose skin diseases of the Dataset-I using Kernel Based ELM.

```
 clc;
clear all;
TrainingData_File=xlsread('H:\Training_70.xlsx');
TestingData_File=xlsread('H:\Testing_30.xlsx');
Elm_Type=1;
NumberofHiddenNeurons=5;
ActivationFunction='sig';
% Kernel_type='poly_kernel';
%Kernel_type='RBF_kernel';
Kernel_type='chisquare_kernel';
Regularization_coefficient=64;
% bestAcc=0;
% for i=-15:15
%  for j=-15:0.05:15
%   Regularization_coefficient=2^i;
```

```
%   kernel_par(1)=2^j;
% [TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy,TY]
= elm_kernel(TrainingData_File, TestingData_File, Elm_Type,
Regularization_coefficient, Kernel_type,kernel_par(1));
%if(TestingAccuracy >= bestAcc)
%bestAcc = TestingAccuracy; bestc
% = Regularization_coefficient; bestg = kernel_par(1);
%end
%end
%end
%fprintf('bestAcc=%f  bestc=%f   bestg=%f ',bestAcc,bestc,bestg)
 TotalTrainingTime=0;
 TotalTestingTime=0;
 TotalTrainingAccuracy=0;
 TotalTestingAccuracy=0;
for i=1:100
[TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy,TY]
 = elm_kernel_1(TrainingData_File, TestingData_File, Elm_Type,
 Regularization_coefficient, Kernel_type,0.98);%RBF kernel
 TotalTrainingTime=TotalTrainingTime+TrainingTime;
 TotalTestingTime=TotalTestingTime+TestingTime;
 TotalTrainingAccuracy=TotalTrainingAccuracy+TrainingAccuracy;
 TotalTestingAccuracy=TotalTestingAccuracy+TestingAccuracy;
end
 AverageTrainingTime = TotalTrainingTime/100
 AverageTestingTime = TotalTestingTime/100
 AverageTrainingAccuracy = TotalTrainingAccuracy/100
```

```
AverageTestingAccuracy = TotalTestingAccuracy/100
```

- Function file to find accuracy using various kernel functions in ELM

```
function [TrainingTime, TestingTime, TrainingAccuracy,
TestingAccuracy,TY]  = elm_kernel(TrainingData_File,
TestingData_File, Elm_Type,  Regularization_coefficient,
Kernel_type, Kernel_para)
% Input:
% TrainingData_File- Filename of training data set
% TestingData_File- Filename of testing data set
% Elm_Type - 0 for regression;
%1 for (both binary and multi-classes) classification
%Regularization_coefficient - Regularization coefficient C
% Kernel_type- Type of Kernels:
%  'RBF_kernel' for RBF Kernel
%  'lin_kernel' for Linear Kernel
%  'poly_kernel' for Polynomial Kernel
%  'wav_kernel' for Wavelet Kernel
%Kernel_para - A number or vector of Kernel Parameters.
% Output:
% TrainingTime-Time(seconds) spent on training ELM
% TestingTime- Time(seconds) spent on predicting ALL testing data
% TrainingAccuracy - Training accuracy:
% RMSE for regression or
%correct classification rate for classification
% TestingAccuracy - Testing accuracy:
```

```
%  RMSE for regression or correct classification rate
% for  Multi-Class Classification:
%Number of output neurons will be
%automatically set equal to number of classes.
% if there are 7 classes in all, there will have 7 output
% Neurons;
%neuron 5 has the highest output means input belongs to 5th class
% Sample1 regression:
% [TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy]
% = elm_kernel('sinc_train','sinc_test',0,1,'RBF_kernel',100)
% Sample2 classification:
%elm_kernel('diabetes_train','diabetes_test',1,1,'RBF_kernel',100)
%===========================Macro definition
REGRESSION=0;
CLASSIFIER=1;
%============= Load training dataset
% train_data=load(TrainingData_File);
% T=train_data(:,1)';
% P=train_data(:,2:size(train_data,2))';
T=TrainingData_File(:,1)';
P=TrainingData_File(:,2:size(TrainingData_File,2))';
clear TrainingData_File;% Release raw training data array
%======= Load testing dataset
TV.T=TestingData_File(:,1)';
TV.P=TestingData_File(:,2:size(TestingData_File,2))';
clear TestingData_File;%Release raw testing data array
C = Regularization_coefficient;
```

```
NumberofTrainingData=size(P,2);
NumberofTestingData=size(TV.P,2);
if Elm_Type~=REGRESSION
%Preprocessing the data of classification
sorted_target=sort(cat(2,T,TV.T),2);
label=zeros(1,1);
%Find and save in 'label' class label from
%training and testing data sets
  label(1,1)=sorted_target(1,1);
  j=1;
for i = 2:(NumberofTrainingData+NumberofTestingData)
  if sorted_target(1,i) ~= label(1,j)
     j=j+1;
    label(1,j) = sorted_target(1,i);
  end
end
 number_class=j;
 NumberofOutputNeurons=number_class;
%=====Processing the targets of training
temp_T=zeros(NumberofOutputNeurons, NumberofTrainingData);
for i = 1:NumberofTrainingData
 for j = 1:number_class
   if label(1,j) == T(1,i)
      break;
   end
 end
temp_T(j,i)=1;
```

```
end
T=temp_T*2-1;
%=======Processing the targets of testing
temp_TV_T=zeros(NumberofOutputNeurons, NumberofTestingData);
for i = 1:NumberofTestingData
  for j = 1:number_class
    if label(1,j) == TV.T(1,i)
    break;
    end
  end
temp_TV_T(j,i)=1;
end
TV.T=temp_TV_T*2-1;
%==end if of Elm_Type
end
%===== Training Phase============================
tic;
n = size(T,2);
Omega_train = kernel_matrix(P',Kernel_type, Kernel_para);
OutputWeight=((Omega_train+speye(n)/C)\(T'));
% size of outputweight=[no_training_sample,no_classes]
 %%a\b meand b is divisible by a and a/b means a divided by b
TrainingTime=toc;
%======= Calculate the training output
Y=(Omega_train * OutputWeight)';
%Y: the actual output of the training data
```

```
%==== Calculate the output of testing input
tic;
Omega_test = kernel_matrix(P',Kernel_type, Kernel_para,TV.P');
TY=(Omega_test' * OutputWeight)';
% TY: the actual output of the testing data
TestingTime=toc;
%Calculate training & testing classification accuracy
if Elm_Type == CLASSIFIER
%Calculate training & testing classification accuracy
  MissClassificationRate_Training=0;
  MissClassificationRate_Testing=0;
for i = 1 : size(T, 2)
  [x, label_index_expected]=max(T(:,i));
  [x, label_index_actual]=max(Y(:,i));
  if label_index_actual~=label_index_expected
  MissClassificationRate_Training=MissClassificationRate_Training+1;
end
end
 TrainingAccuracy=1-MissClassificationRate_Training/size(T,2)
 for i = 1 : size(TV.T, 2)
   [x, label_index_expected]=max(TV.T(:,i));
   [x, label_index_actual]=max(TY(:,i));
   if label_index_actual~=label_index_expected
   MissClassificationRate_Testing=MissClassificationRate_Testing+1;
 end
end
TestingAccuracy=1-MissClassificationRate_Testing/size(TV.T,2)
```

```
end
end

%=== Kernel Matrix

function omega = kernel_matrix(Xtrain,kernel_type, kernel_pars,Xt)
nb_data = size(Xtrain,1);
if strcmp(kernel_type,'RBF_kernel'),
 if nargin<4,% nargin: number of function input parameters
    XXh = sum(Xtrain.^2,2)*ones(1,nb_data);
    omega = XXh+XXh'-2*(Xtrain*Xtrain');
    omega = exp(-omega./kernel_pars(1));
 else
    XXh1 = sum(Xtrain.^2,2)*ones(1,size(Xt,1));
    XXh2 = sum(Xt.^2,2)*ones(1,nb_data);
    omega = XXh1+XXh2' - 2*Xtrain*Xt';
    omega = exp(-omega./kernel_pars(1));
 end
elseif strcmp(kernel_type,'poly_kernel')
 if nargin<4,
  omega = (Xtrain*Xtrain'+kernel_pars(1)).^kernel_pars(2);
 else
  omega = (Xtrain*Xt'+kernel_pars(1)).^kernel_pars(2);
 end
else  %if strcmp(kernel_type,'chisquare_kernel')
 if nargin<4,
  kernel = zeros(size(Xtrain,1),size(Xtrain,1));
```

```
for i=1:size(Xtrain,1)
  d = bsxfun(@minus, Xtrain, Xtrain(i,:));
  s = bsxfun(@plus, Xtrain, Xtrain(i,:));
  kernel(:,i) = sum(d.^2 ./ (s/(0.5)+eps), 2);
end
omega= exp( -kernel_pars(1)* kernel);
 else
  kernel = zeros(size(Xtrain,1),size(Xt,1));
for i=1:size(Xt,1)
  d = bsxfun(@minus, Xtrain, Xt(i,:));
  s = bsxfun(@plus, Xtrain, Xt(i,:));
  kernel(:,i) = sum(d.^2 ./ (s/(0.5)+eps), 2);
end
omega= exp( -kernel_pars(1)* kernel);
end
end
end
```