Table of Contents

Load data

```
clear
clc

filename = 'heartstatlog_trainSet.txt';
delimiterIn = '\t';
Xtrain = importdata(filename,delimiterIn);

filename = 'heartstatlog_trainLabels.txt';
delimiterIn = '\t';
Ytrain = importdata(filename,delimiterIn);

filename = 'heartstatlog_testSet.txt';
delimiterIn = '\t';
Xtest = importdata(filename,delimiterIn);

filename = 'heartstatlog_testLabels.txt';
delimiterIn = '\t';
Ytest = importdata(filename,delimiterIn);
```

Normalization

```
C = bsxfun(@minus,Xtrain,mean(Xtrain));
Xtrain = C/std(Xtrain);

C = bsxfun(@minus,Xtest,mean(Xtest));
Xtest = C/std(Xtest);

Ytrain(Ytrain==2)=-1;
Ytest(Ytest==2)=-1;
```

Kernel matrix (Quadratic)

```
Ktrain = (Xtrain*Xtrain' + 1).^2;
Ktest = (Xtrain*Xtest' + 1).^2;

% Ktrain = (x*y' + 1); % (Linear)
% for i = 1:size(Xtrain,1)
%     for i = 1:size(Xtrain,1)
%         Ktrain = exp(-norm(Xtrain(i,:)-Xtrain(j,:))^2 /(2)); %
 Radial
%         Ktrain = (-norm(Xtrain(i,:)-Xtrain(j,:))/(2)); % Exponential
%     end
% end
```

Alpha

```
lambda = 5;
fun = @(alpha)sum(sum((alpha*alpha').*(Ytrain*Ytrain').*Ktrain
 + lambda*sum((alpha/lambda).*log(alpha/lambda) + (1 - alpha/
lambda).*(log(1 - alpha/lambda)))));
x0 = (ones(size(Ytrain,1),1));
Aeq = Ytrain';
beq = 0;
ub = (1/lambda)*ones(size(Ytrain,1),1);
lb = zeros(size(Ytrain,1),1);
alphas = fmincon(fun,x0,[],[],Aeq,beq,lb,ub,[]);
```

```
Solver stopped prematurely.

fmincon stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 3000 (the default value).
```

b

```
funb = @(b) sum(log(1 + exp(-Ytrain'.*((alphas.*Ytrain)'*Ktrain +
 b))));
b = fminunc(funb,0);
```

```
Warning: Gradient must be provided for trust-region algorithm; using
quasi-newton algorithm instead.

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.
```

Prediction

```
pred = sum((alphas*Ytrain')*Ktest + b);
preds = sign(pred);
error = sum(preds~=Ytest');
```

Published with MATLAB® R2016a