Take Home Exam 6: exercise 2

Alessandro La Farciola

January 3, 2023

In this file there are the Matlab codes for *Exercises (i), (ii), (iii)*.

Exercise (i)

Here it is the code for the solution of part (i).

Firstly, there is the code for *squared-exponential* kernel.

```
function [cov,dl,dsigma]=cov_sqrdexp(x,y,param)
%input:
% param(1)= l
% param(2)= sigma

%output
%cov=covariance matrix
%dl=partial derivative of kernel wrt to param(1)
%dsigma=partial derivative of kernel wrt to param(2)

cov=(param(2)^2)*exp(-(pdist2(x,y).^2)/(2*(param(1)^2)));
if isequal(x,y)
    dl= cov .* ((pdist2(x,y).^2) /(param(1)^3));
    dsigma=2*cov / param(2) ;
else
    warning('x and y are not the same: dl and dsigma are NaN');
    dl=NaN;
   dsigma=NaN;
end
end
```

Secondly, there is the code for *exponential* kernel.

```
function [cov,dl,dsigma]=cov_exp(x,y,param)
%input:
% param(1)= l
% param(2)= sigma

%output
%cov=covariance matrix
%dl=partial derivative of kernel wrt to param(1)
```

```
%dsigma=partial derivative of kernel wrt to param(2)


cov=(param(2)^2)*exp(-(pdist2(x,y))/(2*(param(1)^2)));
if isequal(x,y)
     dl= cov .* ((pdist2(x,y)) /(param(1)^3));
    dsigma=2*cov / param(2) ;
else
    warning('x and y are not the same: dl and dsigma are NaN');
    dl=NaN;
    dsigma=NaN;
end

end
```

Exercise (ii)

Here it is the function *fitGPR* for the solution of part (ii).

```
function [lopt,sigmaopt,post_mean,post_cov]=fitGPR(X,y,KernelFun,sigma0,
    l0,sigma_n,Xtest)

%input
%(X,y)= training data
%KernelFun= kernel function
%sigma0,l0=initial parameters
%Xtest= testing data

%output
%lopt= optimal l
%sigmaopt= optima sigma
%post_mean= posterior mean at testing data
%post_cov= posterior co-variance at testing data


myfun = @(x) maxl(KernelFun, X, y, sigma_n, x(1),x(2));

options = optimset('GradObj','on');

x = fminlbfgs(myfun, [sigma0, l0],options);
sigmaopt=x(1);
lopt=x(2);

Ky_ = KernelFun(X,X,[lopt sigmaopt]) + sigma_n*eye(length(y));
Kstar = KernelFun(X,Xtest,[lopt sigmaopt]);
Kstarstar = KernelFun(Xtest,Xtest,[lopt sigmaopt]);
L_ = chol(Ky_, "lower" );
alpha_ = L_'\(L_\y);
post_mean = Kstar' * alpha_;
```

```
v=L_\Kstar;
post_cov= Kstarstar-v'*v;
end

function [loglike, grad] = maxl(KernelFun, X, y,sigma_n,sigma,l)
[Kernel,dKl,dKsigma] = KernelFun(X,X,[l sigma]);
n = length(y);
Ky = Kernel + eye(n) * (sigma_n^2);
L = chol(Ky, "lower" );
alpha = L'\(L\y);
logdet = 2*sum(log( diag( L )));
loglike = - (-y' * alpha./2 - logdet./2 - n*log(2*pi)/2);
grad_sigma = -(0.5* trace( (alpha * alpha')*dKsigma - L'\(L\dKsigma)) );
grad_l = -( 0.5* trace( (alpha * alpha')*dKl - L'\(L\dKl)));
grad = [grad_sigma grad_l];
end
```

Exercise (iii)

In this section, I firstly present the script for part (iii).

```
r=0.001;
K=130;
sigma=0.1;
T=1.0;

n_train=100; % number of training instances
n_test=50; % number of test instances

S_train=50+(200-50)*unifrnd(0,1,n_train,1);
S_test=50+(200-50)*unifrnd(0,1,n_test,1);

[price_train,~]=blsprice(S_train,K,r,T,sigma);
[price_test,~]=blsprice(S_test,K,r,T,sigma);

% Predict option prices with fitGPR of part (ii)
%take as sigman the optimal according to fitgpr of MATLAB
gprMdl1 = fitrgp(S_train,price_train,'BasisFunction','none','KernelFunction','squaredexponential');
sigman=gprMdl1.Sigma;
[lopt,sigmaopt,post_mean,post_cov]=fitGPR(S_train,price_train,@cov_sqrdexp,10, 10, sigman ,S_test);
opt_param=[lopt sigmaopt];

figure
plot(price_test,post_mean)
xlabel("BS prices")
ylabel("Predicted prices")

```

```
aae_prices=mean(abs(price_test-post_mean))
mae_prices=max(abs(price_test-post_mean))

% true values of delta
[delta_test,~]=blsdelta(S_test,K,r,T,sigma);

% predicting delta values with fitGPR
[cov1,~]=cov_sqrdexp(S_train,S_train,opt_param);
R=chol(cov1+(sigman^2)*eye(length(S_train)));
alpha=R\(R'\price_train);
[cov,~]=cov_sqrdexp(S_test,S_train,opt_param);
delta_predicted=(-1/(opt_param(1)^2))*((S_test-S_train').*cov)*alpha;

figure
plot(delta_test,delta_predicted)
xlabel("Delta")
ylabel("Predicted delta")
aae_delta=mean(abs(delta_test-delta_predicted))
mae_delta=max(abs(delta_test-delta_predicted))
```

The following plots show the results obtained.

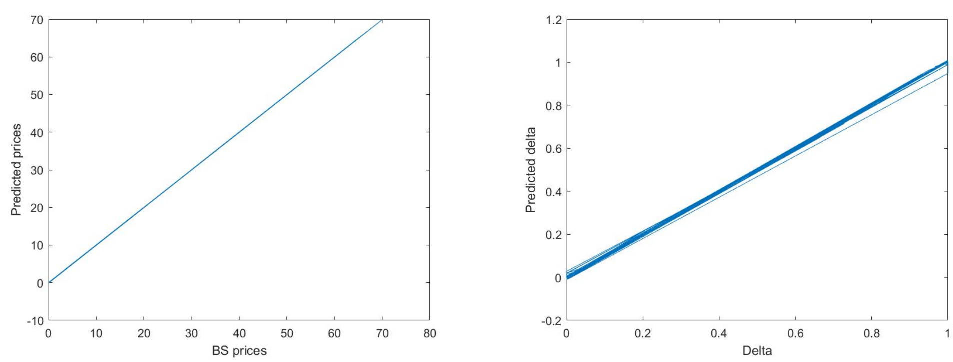

Figure 1: Exact price vs predicted price and exact delta vs predicted delta

Moreover, in the following table it is possible to see the maximum absolute error (MAE) and the average absolute error (AAE) for both predictions.

	Price	*Delta*
MAE	0.1138	0.0518
AAE	0.0447	0.0093

Table 1: Errors.