

```

classdef KernelRegressionClass<handle
% Clase para ejemplos de regresión Kernel semiparametrica.
% (c) 2022, Antonio Sala Piqueras. Universitat Politecnica de Valencia. Todos los derechos
reservados.
% Presentación video funcionalidad de la clase: http://personales.upv.es/asala/YT/V/semipcls.html
% Teoría en: http://personales.upv.es/asala/YT/V/semipar1.html y http://personales.upv.es/asala/YT/V/semipar2.html
% Caso de estudio utilizacion: http://personales.upv.es/asala/YT/V/semipcaso.html

    properties
        M %varianza componente Kernel
        sg %param desv. kernel gaussiano
        ThCV %componente parametrico, vza inicial
        lam %ruido medida
        phi %comp. paramétrico
        data %X, y
        result %datos de resultados
    end

    methods
        function obj = KernelRegressionClass(M,sg,ThCV,lam)
            obj.M=M;obj.sg=sg;obj.ThCV=ThCV;obj.lam=lam;
            if isempty(ThCV)
                obj.phi=[]; %sin componente paramétrico
            end
        end

        function ke=KernelMatr(obj,x,X,solodiag)
            arguments
                obj
                x
                X
                solodiag="full"
            end
            M=obj.M;sg=obj.sg;
            kappa=@(x1,x2) M*exp(-norm(x2-x1)^2/2/sg^2); %Gaussian Kernel
            nx=size(x,1); nX=size(X,1);
            ke=zeros(nx,nX);
            indiceJ=1:nX;
            for i=1:nx
                if(solodiag=="diag") %solo queremos calcular varianzas a priori
                    indiceJ=i;
                end
                for j=indiceJ
                    ke(i,j)=kappa(x(i,:),X(j,:));
                end
            end
            if(solodiag=="diag")
                ke=diag(ke); %se puede mejorar sin pedir matriz entera si solo quiero
                diagonal, pero abrevio.
            end
        end

        function DoKernelRegression(obj,X,y)
            obj.data.X=X;
            obj.data.y=y;
            obj.result.Knparam=obj.KernelMatr(X,X);
            N=size(X,1);
            if isempty(obj.phi)
                obj.result.Ktot=obj.result.Knparam;
            else
                ophi=obj.phi(X);
                obj.result.Ktot=obj.result.Knparam+ophi*obj.ThCV*ophi';
            end
        end
    end
end

```

```

        obj.result.phiX=phi;
    end
    obj.result.invKtot=inv(obj.result.Ktot+obj.lam*eye(N));
    obj.result.w=obj.result.invKtot*y;
    if(~isempty(obj.phi))
        obj.result.th=obj.ThCV*phi'*obj.result.w; %estimado de los parámetros.
        obj.result.varth=obj.ThCV-obj.ThCV*phi'*obj.result.invKtot*phi*obj.ThCV;
        obj.result.stdth=sqrt(diag(obj.result.varth));
    end
end

function [ymean, dy_desvtip]= Predict(obj,Xtest)
    obj.data.Xtest=Xtest;
    X=obj.data.X;
    KtestX=obj.KernelMatr(Xtest,X);
    obj.result.varpriortest=obj.KernelMatr(Xtest,Xtest,'diag');
    obj.result.componenteNoParametrico=KtestX*obj.result.w;
    if(isempty(obj.phi))
        ymean=obj.result.componenteNoParametrico;
        Ktot_test_X=KtestX;
    else
        phi=obj.phi(Xtest);
        obj.result.varpriortest=obj.result.varpriortest+phi*obj.ThCV*phi';
        obj.result.componenteParametrico=phi*obj.result.th;
        Ktot_test_X=KtestX+phi*obj.ThCV*phi';
        ymean=obj.result.componenteNoParametrico+obj.result.componenteParametrico;
    end
    yvzT=obj.result.varpriortest-Ktot_test_X*obj.result.invKtot*Ktot_test_X';
    dy_desvtip=sqrt(diag(yvzT));
    obj.result.ymean=ymean;
    obj.result.desvtip=dy_desvtip;
end

function PlotTrainData(obj)
    plot(obj.data.X,obj.data.y,'xk','LineWidth',3);
end

function PlotFittedFunction(obj)
    Xtest=obj.data.Xtest;
    ymean=obj.result.ymean;
    dyvzT=obj.result.desvtip;
    plot(Xtest,ymean,'LineWidth',3);
    hold on
    if(~isempty(obj.phi))
        plot(Xtest,obj.result.componenteParametrico,'Color',[.55 .75 .
5], 'LineStyle','-','LineWidth',2);
    end
    plot(Xtest,[ymean+2*dyvzT ymean-2*dyvzT],'Color',[.6 .7 .
75], 'LineStyle',':','LineWidth',2)
    grid on, hold off, axis tight
end

end %Methods
end %classdef

```