CLASS ONLINE-SVR REFERENCE MANUAL

RETURN	METHOD NAME	PARAMETERS	DESCRIPTION
I N I T I A L I Z A T I O N			
	OnlineSVR	void	New OnlineSVR
void	Clear	void	Clear the OnlineSVR
A T T R I B U T E S			
double	GetC	void	Get *C* parameter
void	SetC	double C	Set *C* parameter
double	GetEpsilon	void	Get *Epsilon* parameter
void	SetEpsilon	double Epsilon	Set *Epsilon* parameter
int	GetKernelType	void	Get *KernelType* parameter
void	SetKernelType	int KernelType	Set *KernelType* parameter
double	GetKernelParam	void	Get *KernelParam* parameter
void	SetKernelParam	double KernelParam	Set *KernelParam* parameter
double	GetKernelParam2	void	Get *KernelParam2* parameter
void	SetKernelParam2	double KernelParam2	Set *KernelParam2* parameter
bool	GetAutoErrorTollerance	void	Get *AutoErrorTollerance* parameter
void	SetAutoErrorTollerance	bool AutoErrorTollerance	Set *AutoErrorTollerance* parameter
double	GetErrorTollerance	Void	Get *ErrorTollerance* parameter
void	SetErrorTollerance	double ErrorTollerance	Set *ErrorTollerance* parameter
int	GetVerbosity	void	Get *Verbosity* parameter
void	SetVerbosity	int Verbosity	Set *Verbosity* parameter
bool	GetStabilizedLearning	void	Get *StabilizedLearning* parameter
void	SetStabilizedLearning	bool StabilizedLearning	Set *StabilizedLearning* parameter
int	GetSamplesTrainedNumber	void	Get number of samples trained
int	GetSupportSet ElementsNumber	void	Get number of support set elements
int	GetErrorSet ElementsNumber	void	Get number of error set elements
int	GetRemaningSet ElementsNumber	void	Get number of remaining set elements
Vector<int>*	GetSupportSetIIndexes	void	Get *SupportSetIndexes* list
Vector<int>*	GetErrorSetIndexes	void	Get *ErrorSetIndexes* list
Vector<int>*	GetRemainingSetIndexes	void	Get *RemainingSetIndexes* list
L E A R N I N G M E T H O D S			
int	Train	Matrix<double>* X Vector<double>* Y	Train samples (*X*,*Y*)
int	Train	double** X double* Y int ElementsNumber int ElementsSize	Train samples (*X*,*Y*)
int	Train	Vector<double>* X Vector<double>* Y	Train sample (X,Y)
int	Forget	Vector<int>* Indexes	Forget the samples of position *Indexes*
int	Forget	int* Indexes int N	Forget the *N* samples of position *Indexes*
int	Stabilize	void	Re-train the OnlineSVR until KKT conditions are verified

CLASS ONLINE-SVR REFERENCE GUIDE

RETURN	METHOD NAME	PARAMETERS	DESCRIPTION
P R E D I C T / M A R G I N M E T H O D S			
double	Predict	Vector<double>* X	Predict the value of sample *X*
double	Predict	double* X int ElementsSize	Predict the value of sample *X*
Vector<double>*	Predict	Matrix<double>* X	Predict the value of samples *X*
double*	Predict	double** X int ElementsNumber int ElementsSize	Predict the value of samples *X*
double	Margin	Vector<double>* X double Y	predicted value error of *X* compared with value of *Y*
double	Margin	double* X int ElementsSize double Y	predicted value error of *X* compared with value of *Y*
Vector<double>*	Margin	Matrix<double>* X Vector<double>* Y	predicted values error of *X* compared with values of *Y*
double*	Margin	double** X double* Y int ElementsNumber int ElementsSize	predicted values error of *X* compared with values of *Y*
C O N T R O L M E T H O D S			
bool	VerifyKKTConditions	void	Check if KKT conditions are verified in current OnlineSVR
void	FindError	Matrix<double>* ValidationSetX Vector<double>* ValidationSetY double* MinError double* MeanError double* MaxError	Find errors of a new *ValidationSet* and compute the *MinError*, the *MeanError* and the *MaxError*
I N P U T / O U T P U T M E T H O D S			
void	ShowInfo	void	Show OnlineSVR statistics
void	ShowDetails	void	Show OnlineSVR details
void	LoadOnlineSVR	char* Filename	Load OnlineSVR from a file
void	SaveOnlineSVR	char* Filename	Save OnlineSVR into a file
void	Import	char* Filename Matrix<double>* X Vector<double>* Y	Import new data from a file
void	Import	char* Filename Matrix<double>** AngularPositions Matrix<double>** MotorCurrents Matrix<double>** AppliedVoltages	Import new robot data from a file

CLASS ONLINE-SVR REFERENCE GUIDE

CONSTANT NAME	CONSTANT DESCRIPTION
K E R N E L C O N S T A N T S	
KERNEL_LINEAR	Linear Kernel
KERNEL_POLYNOMIAL	Polynomial Kernel
KERNEL_RBF	Radial Basis Function Kernel
KERNEL_RBF_GAUSSIAN	Gaussian RBF Kernel
KERNEL_RBF_EXPONENTIAL	Exponential RBF Kernel
KERNEL_MLP	MultiLayer Perceptron Kernel
V E R B O S I T Y C O N S T A N T S	
VERBOSITY_NO_MESSAGES	No messages on video
VERBOSITY_NORMAL	Training basic informations
VERBOSITY_DETAILS	Training details
VERBOSITY_DEBUG	Training and variations details

CLASS ONLINE-SVR EXAMPLES

```
#include "OnlineSVR.h"
#include <math.h>

using namespace onlinesvr;


int main ()
{
	// Make a new OnlineSVR
	OnlineSVR* SVR = new OnlineSVR();

	// Set parameters
	SVR->SetC(20);
	SVR->SetEpsilon(0.01);
	SVR->SetKernelType(OnlineSVR::KERNEL_RBF);
	SVR->SetKernelParam(30);
	SVR->SetVerbosity(OnlineSVR::VERBOSITY_NORMAL);

	// Build the training set
	Matrix<double>* TrainingSetX = Matrix<double>::RandMatrix(20,1);
	Vector<double>* TrainingSetY = new Vector<double>();
	for (int i=0; i<TrainingSetX->GetLengthRows(); i++)
		TrainingSetY->Add(sin(TrainingSetX->GetValue(i,0)));

	// Train OnlineSVR
	SVR->Train(TrainingSetX,TrainingSetY);

	// Show OnlineSVR info
	SVR->ShowInfo();

	// Predict some new values
	Matrix<double>* TestSetX = new Matrix<double>();
	Vector<double>* X1 = new Vector<double>();
	Vector<double>* X2 = new Vector<double>();
	X1->Add(0); TestSetX->AddRowRef(X1);
	X2->Add(1); TestSetX->AddRowRef(X2);
	Vector<double>* PredictedY = SVR->Predict(TestSetX);
	cout << "f(0) = " << PredictedY->GetValue(0) << endl;
	cout << "f(1) = " << PredictedY->GetValue(1) << endl;

	// Forget some samples
	Vector<int>* RemainingSamples = SVR->GetRemainingSetIndexes()->Clone();
	SVR->Forget(RemainingSamples);

	// Save OnlineSVR
	SVR->SaveOnlineSVR("Sin.svr");

	// Delete
	delete SVR;
	delete TrainingSetX;
	delete TrainingSetY;
	delete TestSetX;
	delete PredictedY;
	delete RemainingSamples;
}
```