Natural & High Performance Computing Laboratory
NHC Lab
http://atm.neuro.pub.ro/radu_d/

Comparative performances of XFSVC and SVM for IRIS, SATIMG, PHONEME and OPTD64 datasets

R. Dogaru and Ioana Dogaru ,
NHC Lab,
May 25, 2016.

Platform: Octave 4.0 on Windows-XP, Intel 2-core E7500 2.93Ghz

SVM – LIBSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvm/)
implementation (.MEX prepared for Octave)

FSVC-NT (no-tune – fastest – version of FSVC, with eta=0); This version gives a large number of RBF units but using fast training and with some advantages in embedded systems design.

FSVC with tuned Adaline (standard version of FSVC) – may achieve better performance with less units but larger training times.

IRIS problem

Implement	Parameters	RBF-units (or SVs)	Training time (s)	Accuracy	Notes
FSVC-NT	xfsvc(1,.6,0/16,'rbf_dog','manh','iris',20); xfsvc(.1,.5,0/16,'rbf_dog','manh','iris',20);	15 7	0[1]	**100%**	
SVM	Gaussian kernel $(\gamma, C) = (0.1,10)$	27	0	94%	

PHONEME problem

Implement	Parameters	RBF-units (or SVs)	Training time (s)	Accuracy	Notes
FSVC-NT	xfsvc(4,0.14,0,'rbf_dog','manh','phoneme',2); xfsvc(1,0.03,0,'rbf_gus','eucl','phoneme',2);	1565 2329	0.093 0.36	87.23% **88.82%**	
SVM	Gaussian kernel $(\gamma, C) = (28,10)$	961	0.468	88.675%	

[1] Too small to be measured with routines from time.h

OPTD64 problem

Implement	Parameters	RBF-units (or SVs)	Training time (s)	Accuracy	Notes
FSVC-NT	xfsvc(.25,1,0,'rbf_gus','eucl','optd64',2); xfsvc(1,7.6,0,'rbf_dog','manh','optd64',2);	1720 595	0.953 0.218	**98.664%** 97.663%	
SVM	Gaussian kernel $(\gamma, C) = (0.046,10)$	1076	0.95	98.442%	

SATIMG problem

Implement	Parameters	RBF-units (or SVs)	Training time (s)	Accuracy	Notes
FSVC-NT	xfsvc(.5,.15,0,'rbf_gus','eucl','satimg',2);	2115	0.6	91.67%	
FSVC	xfsvc(4,.25,1/32,'rbf_gus','eucl','satimg',8);	1654	2.06	**91.889%**	
SVM	Gaussian kernel $(\gamma, C) = (1.76,10)$	1244	0.75	91.67%	

Usage:

- Data should be prepared using the LIBSVM format;
- Data should be randomized (consecutive samples from random classes in the training set)
- Tuning advice: Start with "prag=1" and a big radius (e.g. 64), then divide by 2 the value until entering a region with good accuracy; fine tuning of radius (and eventually the "prag" parameters) until getting the best performance
- More details in the list of papers from xfsvc.m (to be cited when used in published work)
- Details on parameters and running examples are included in xfsvc.m file