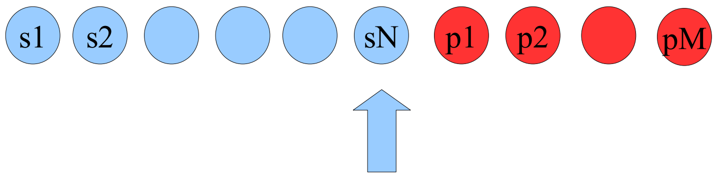
SMF Stock Market Forecaster 1.0

About SMF Tool

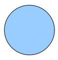
We have developed an efficient tool for intraday stock market forecasting based on **Neural Networks** and **Wavelet Decomposition**. This software has been tested on real data obtaining excellent results. SMF Tool gives Buy/Sell signals with a high degree of accuracy.

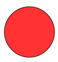
How SMF works

SMF accepts, as input, a sequence of given length N. The system can determine if at least one of future prices - within an observation window of fixed length M - will be higher or lower than current price.



Current price

 Element of input sequence

 Element of predicted sequence

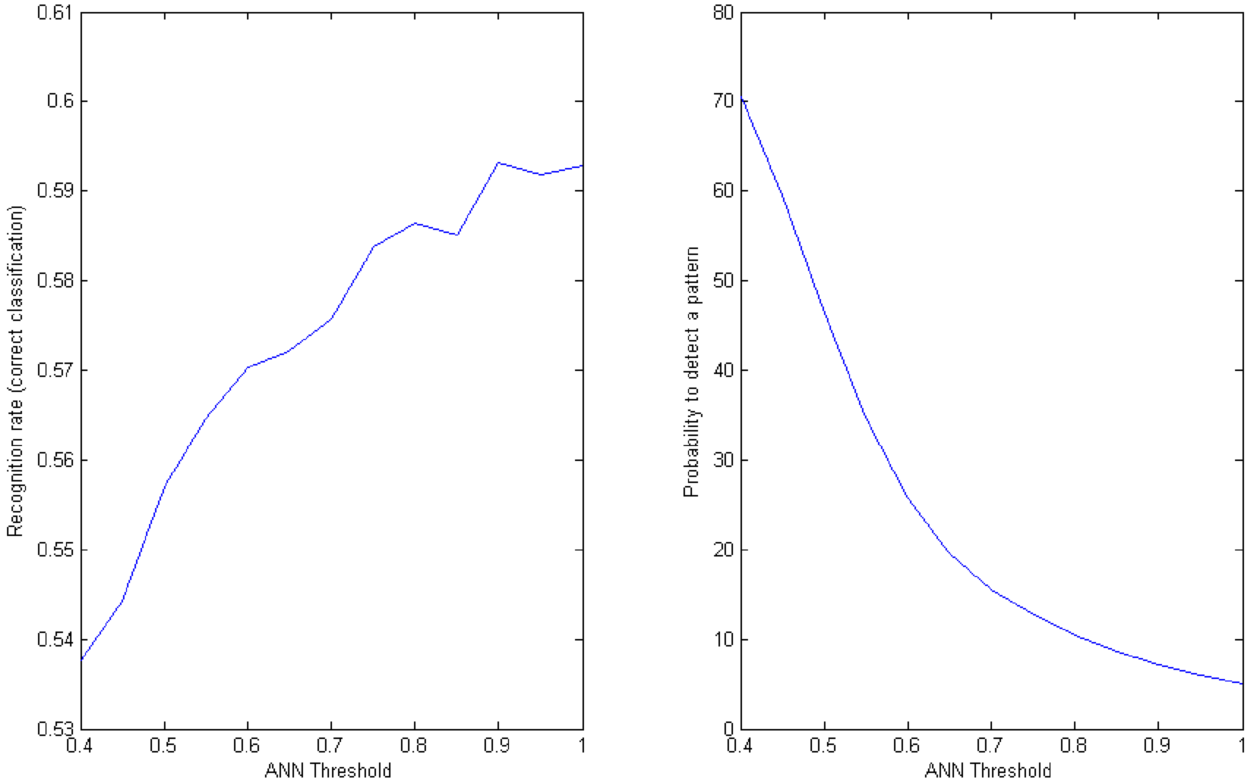
A **buy signal** is given if at least one of red circles is greater than current price. A **sell signal** is given if at least one of red circles is lower than current price. Formally a buy signal is verified if (p1>sN) OR (p2>sN) OR ... OR (pM>sN), a sell signal is verified if (p1<sN) OR (p2<sN) OR ... OR (pM<sN).

Performances

SMF package has been tested with Italian Futures over a period of 3 years, more than 600 days of effective trading. Training data and testing data have been randomly selected from this data set, without any overlapping. Stock market data have been downloaded at http://www.ccg.it : these data are uniformly sampled each minute.

	SMF Classifier	**No Classifier**
Test #1: N=32, M=3, **Buy**	0.5751	0.5312
Test #2: N=32, M=3, **Buy**	0.5791	0.5319
Test #3: N=32, M=4, **Buy**	0.6048	0.5785
Test #4: N=32, M=4, **Buy**	0.6087	0.5887
Test #5: N=32, M=5, **Buy**	0.6476	0.6231
Test #6: N=32, M=5, **Buy**	0.6593	0.6216
Test #7: N=32, M=3, **Sell**	0.5602	0.5200
Test #8: N=32, M=3, **Sell**	0.5660	0.5228
Test #9: N=32, M=4, **Sell**	0.6068	0.5686
Test #10: N=32, M=4, **Sell**	0.6099	0.5761
Test #11: N=32, M=5, **Sell**	0.6543	0.6146
Test #12: N=32, M=5, **Sell**	0.6305	0.6089

The table above shows the probability that in the observation window of length M there is at least one value higher (buy signal) or lower (sell signal) than current price using the SMF classifier and without using any classifier.



The image above shows how system performances can be improved using a higher threshold for nonlinear classifier output. Increasing this threshold value we achieve a better recognition rate (i.e. # correct detected patterns / # total detected patterns) but the number of Buy/Sell signals (i.e. # total detected patterns / # total input sequences) decreases.

Why Wavelets

Wavelets can localize data in time-scale space. At high scales (shorter time intervals), the wavelet has a small time support and is thus, better able to focus on short lived, strong transients like discontinuities, ruptures and singularities. At low scales (longer time intervals), the wavelet's time support is large, making it suited for identifying long periodic features. Wavelets have a intuitive way of characterizing the physical properties of the data. At low scales, the wavelet characterizes the data's coarse structure; its long-run trend and pattern. By gradually increasing the scale, the wavelet begins to reveal more and more of the data's details, zooming in on its behavior at a point in time. Wavelet analysis is the analysis of change. A wavelet coefficient measures the amount of information that is gained by increasing the frequency at which the data is sampled, or what needs to be added to the data in order for it to look like it had been measured more frequently. For instance, if a stock price does not change during the course of a week, the wavelet coefficients from the daily scale are all zero during that week. Wavelet coefficient that are non-zero at high scales typically characterize the noise inherent in the data. Only those wavelets at very fine scales will try to follow the noise, whereas those wavelets at coarser scales are unable to pick up the high frequency nature of the noise.

Why Neural Networks

Since the early 90's when the first practically usable types emerged, artificial neural networks (ANNs) have rapidly grown in popularity. They are artificial intelligence adaptive software systems that have been inspired by how biological neural networks work. Their use comes in because they can learn to detect complex patterns in data. In mathematical terms, they are universal non-linear function approximators meaning that given the right data and configured correctly, they can capture and model any input-output relationships. This not only removes the need for human interpretation of charts or the series of rules for generating entry/exit signals but also provides a bridge to fundamental analysis as that type of data can be used as input. In addition, as ANNs are essentially non-linear statistical models, their accuracy and prediction capabilities can be both mathematically and empirically tested. In various studies neural networks used for generating trading signals have significantly outperformed buy-hold strategies as well as traditional linear technical analysis methods. While the advanced mathematical nature of such adaptive systems have kept neural networks for financial analysis mostly within academic research circles, in recent years more user friendly neural network software has made the technology more accessible to traders.

Quick start

Unzip all files in Matlab current directory and type “test_main” on Matlab command window.

Requirements:

Matlab, Matlab Image Processing Toolbox, Matlab Neural Network Toolbox, Matlab Wavelet Toolbox.

In order to obtain the complete source code please visit

http://www.advancedsourcecode.com/neuralnetworkforecasting.asp

Luigi Rosa
Via Centrale 35
67042 Civita Di Bagno
L'Aquila – ITALY
Mobile +39 3207214179
Email luigi.rosa@tiscali.it
Web site http://www.advancedsourcecode.com