A Kernel-Based Technique for Direction-of-Change Financial Time Series Forecasting

Andrew Skabar

Department of Computer Science and Computer Engineering
La Trobe University, Victoria 3086 Australia
a.skabar@latrobe.edu.au

Abstract. This paper presents a generative approach to direction-of-change time series forecasting. Kernel methods are used to estimate densities for the distribution of positive and negative returns, and these distributions are then combined to produce probability estimates for return forecasts. An advantage of the technique is that it involves very few parameters compared to regression-based approaches, the only free parameters being those that control the shape of the windowing kernel. A special form is proposed for the kernel covariance matrix. This allows recent data more influence than less recent data in determining the densities, and is important in preventing overfitting. The technique is applied to predicting the direction of change on the Australian All Ordinaries Index over a 15 year out-of-sample period.

Key words: Financial time series forecasting, kernel methods

1 Introduction

Financial time series forecasting has almost invariably been approached as an autoregression problem in which future values of a time series are predicted on the basis of past values. The parameters of the prediction model are first optimized using in-sample data; the model is then used to make forecasts on a set of out-of-sample data, and the accuracy of the forecasts is measured by comparing the forecast values with the realized (i.e., actual) values.

In evaluating forecast accuracy, the mainstream literature has tended to focus almost exclusively on measuring error magnitudes. For example, in their review of time series forecasting covering a 25 year period, De Gooijer and Hyndman (2006) include a section on forecast evaluation and accuracy measures in which they list 17 commonly used forecast accuracy measures, all of which are based on the magnitude of the error between the predicted and realized values [1]. This is interesting because in many cases the magnitude of the error is not as important as whether the direction of the prediction (i.e., *up* or *down*) is correct. For example, if one is going to make stock trading decisions based on forecast values, it is more important to be able to correctly predict the direction of change than it is to achieve, say, a small mean squared error. We call this *direction-of-change forecasting*. The importance of correctly predicting the direction of change has been acknowledged in several recent papers [2-5].

M. Bubak et al. (Eds.): ICCS 2008, Part II, LNCS 5102, pp. 441 – 449, 2008.

One approach to direction-of-change forecasting is simply to use a regression model to forecast the next value of the time series, and to then convert this to a direction prediction by comparing the forecast value with the current value: if the forecast value is greater than the current value of the series, predict *up*; otherwise predict *down*. That is, direction-of-change forecasting can be seen as simply a regression problem involving this extra step.

Many regression models have been proposed and tested over the years. Many of these models are linear, but in recent years non-linear models have also become popular. For example, there are many reports in the literature of the application of neural networks to financial time series forecasting [2][6-8]. There is, however, considerable debate about whether non-linear models are able to provide better forecasts than linear or random walk models when applied to financial time-series data [9-11]. One of the main problems with non-linear models such as neural networks is that they involve a large number of parameters. A typical neural network architecture will have in the order of 100s of weights which must be optimized, usually using some gradient-descent based technique. This can lead to severe over-fitting problems, especially on noisy data such as financial commodity prices. Preventing over-fitting requires careful selection of number of hidden units, regularization coefficients, and early stopping point, and these usually require an expensive cross-validation procedure. Furthermore, there is no guarantee that training from a different set of initial weights will results in the same predictions.

In this paper we take a conceptually different approach to direction-of-change forecasting. Rather than treating the problem as a regression problem, we conceptualize the problem as a binary classification problem, and predict the direction of change directly. There are several reasons as to why such an approach might be advantageous. Firstly, traders base their trading decisions primarily on their opinion of whether the price of a commodity will rise or fall, and to a lesser extent on their opinion of the degree with which it will rise or fall. This may create in financial systems an underlying dynamic that allows the direction of change to be predicted more reliably than the actual value of the series. Secondly, conceptualizing the problem as a classification problem allows us to apply a different family of algorithms. In this paper, we are specifically interested in applying generative classification models. That is, rather than discovering a function which maps directly from a set of inputs onto a prediction of the direction of change (as would be the case with a neural network approach, for example), the approach we present in this paper is based on estimating probability densities and combining these under Bayes' Theorem to arrive at *a posteriori* estimates of the probabilities of upward and downward movements. The main advantage of this approach over discriminative approaches is that it leads to more parsimonious models that involve few parameters.

The remainder of the paper is structured as follows. Section 2 reviews the basic framework for density estimation-based classification and outlines how this may be applied to direction-of-change time series forecasting. Section 3 describes the kernel covariance matrix parameterization, together with a cross-validation procedure for optimizing the covariance matrix parameters. Section 4 presents and discusses empirical results of applying the techniques to the Australian All Ordinaries (AORD) Index. Section 5 concludes the paper.

2 Generative Models for Classification

Generative classification models are based on estimating the probability distributions from which the data was generated. Typically, the probability density functions are estimated; these densities are then combined using Bayes' Theorem to produce posterior probabilities; and these probabilities are then used to perform classification. In this section we describe how density estimation-based methods can be applied to direction-of-change time series forecasting.

2.1 Preliminaries

Because financial commodity price series are usually highly non-stationary it is common to apply some type of transformation to the raw price data, thus obtaining transformed variables. Transformed variables are typically based either on absolute or relative changes in price, and in this paper we use *returns*. The return on day t is defined as $r_t = (p_t - p_{t-1}) / p_{t-1}$, where p_t and p_{t-1} are the prices respectively on days t and $t-1$. Our objective is to predict the direction of the change in price on day t; that is, we wish to predict the sign of r_t. More specifically, we wish to predict the sign of r_t on the basis of the returns observed on the D days preceding day t. That is, we wish to predict sign(r_t) on the basis of the vector ($r_{t-1}, r_{t-2}, \ldots , r_{t-D}$), which we refer to as the *delayed-return vector* for day t. For notational convenience, we will represent the delayed-return vector as a column vector $\mathbf{x}_t = (r_{t-1}, r_{t-2}, \ldots , r_{t-D})^T$, which we refer to as a *data point*. If there are N data points, then we represent the collection of these using the set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $|\mathbf{X}| = N$.

2.2 Density Estimation

There are three general approaches to explicitly estimating probability densities: the parametric, non-parametric, and semi-parametric (or mixture-model) approaches. The parametric approach assumes the form of the distribution (e.g., Gaussian), and the task is to estimate the values of the parameters for that distribution (e.g., the mean and covariance in the Gaussian case). A problem with this approach is that many datasets do not follow a standard distribution, and attempts to model them in this way may lead to very poor estimates of the distribution.

A second approach—the *kernel,* or *Parzen*, method—is a non-parametric approach that involves modelling the distribution using a series of probability windows (usually Gaussian) centred at each sample [12][13]. The overall probability density function is the average of all of the individual distributions centred at each point: i.e.,

$$p(\mathbf{x}) = \frac{1}{|\mathbf{X}|} \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \sum_{n=1}^{|\mathbf{X}|} \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{x}_n)^T \Sigma^{-1} (\mathbf{x}-\mathbf{x}_n)) \tag{1}$$

where $\mathbf{x}$ is the point at which the density is estimated, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is the set of points from which the density is estimated, the factor $1/(2\pi)^{D/2}|\Sigma|^{1/2}$ is a constant which ensures that the area under each Gaussian sums to one, and Σ is the covariance matrix for the Gaussian kernel. The main decision to be made in using this approach is the selection of Σ, which defines the shape of the windowing kernel function, and acts as the smoothing parameter. If the kernel is too narrow then the distribution will be peaked around each of the sample points; if the kernel is too wide then the distribution will be overly smoothed.

A third approach is the semi-parametric, or *mixture model*, approach, which can be seen as a compromise between the parametric and non-parametric approaches. In this case K distributions are used to model the data, where K is much smaller than the number of sample points. Only the non-parametric approach is used in this paper.

2.3 Classification

Bayes' Theorem states that the posterior probability than an observation **x** belongs to class C_k is given by

$$P(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k) P(C_k)}{p(\mathbf{x})}, \tag{2}$$

where $p(\mathbf{x} \mid C_k)$ is the class-conditional probability density function for examples belonging to class C_k, $P(C_k)$ is the prior probability than an example belongs to class C_k, and can be estimated from the training data, and $p(\mathbf{x})$ is the unconditional probability density function for **x**. Note that $p(\mathbf{x})$ can be determined using

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x} \mid C_k) P(C_k), \tag{3}$$

and can thus be seen as a normalizing factor that ensures that the sum of probabilities over all classes is unity. An example **x** is classified into the class C_k for which $P(C_k \mid \mathbf{x})$ is a maximum.

Direction-of-change forecasting is a binary classification problem in which an example belongs either to the class C_+ (upward movement), or C_- (downward movement). Assuming that the data points **X** are partitioned into two sets, $\mathbf{X}_+$ and **X**, containing respectively data points corresponding to upward and downward movements, then the Parzen estimate for $p(\mathbf{x} \mid C_+)$ is

$$P(\mathbf{x} \mid C_+, \Sigma) = \frac{1}{|\mathbf{X}_+|} \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \sum_{n=1}^{|\mathbf{X}_+|} \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_{+n})^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}_{+n})) \tag{4}$$

with a corresponding expression for $P(\mathbf{x} \mid C_-, \Sigma)$. The posterior probabilities $P(C_+ \mid \mathbf{x}_n, \Sigma)$ and $P(C_- \mid \mathbf{x}_n, \Sigma)$ are estimated using Bayes' Theorem.

3 Covariance Matrix Optimization

In this paper we estimate densities using the non-parametric method described above. Therefore the only parameter to be determined is the covariance matrix defining the

Gaussian kernel used in the density estimation process. We first discuss how the covariance matrix may itself be parameterized, and we then discuss how these parameters may be optimized.

3.1 Covariance Matrix Parameterization

There are three general forms that the covariance matrix Σ may take: *spherical*, in which the variance along each dimension is identical; *diagonal*, in which case the variances along different dimensions differ, but principal directions are aligned with the coordinate axes; and *full*, in which case the principal direction of the variances can be aligned in arbitrary directions. Because Σ is a symmetric matrix, in the most general case it has $D(D+1)/2$ independent components, where D is the dimensionality of the input space (i.e., the number of delayed returns). In the diagonal case all of the non-diagonal elements of the covariance matrix are zero, and thus the number of independent components of the covariance matrix reduces to D. In the spherical case the diagonal components of the covariance matrix are all equal, and hence there is only one non-zero component [14].

Data points in a time series are temporally ordered, and intuitively we would expect recent values of the time series to be more important than less recent values in predicting future values; that is, we expect the prediction to be more sensitive to recent values. This suggests that the kernel used to estimate a density should not be symmetrical (i.e., spherical), but of a form such that its width along dimensions corresponding to recent returns is smaller than its width along dimensions corresponding to less recent returns. To capture the requirement that the width of the kernel increases with the delay of the return, we use the scaling

$$\Sigma_{t-n} = a\,\mathrm{e}^{k(n-1)}, \tag{5}$$

where Σ_{t-n} ($n \in \{1, 2, ..., D\}$) is the variance of the kernel in the direction parallel to the axis corresponding to delay r_{t-n}, k is an exponential scaling factor, and a is the variance parallel to the first delay[1]. Thus the kernel covariance matrix $\mathbf{\Sigma}$ has the form:

$$\begin{bmatrix} a & 0 & 0 & \cdots & 0 \\ 0 & ae^{k} & 0 & \cdots & 0 \\ 0 & 0 & ae^{2k} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & ae^{(D-1)k} \end{bmatrix}.$$

Note that this form of covariance reduces sensitivity to the value of D, as additional returns will have increasingly less influence on the estimate of the density.

[1] In our experiments we have defined a slightly differently as $\Sigma_{t-n} = a \cdot v \cdot \mathrm{e}^{k(n-1)}$, where v is the variance of the returns in the training period. This rescaling makes a less variable across different datasets, and makes parameter optimization easier.

3.2 MAP Optimization of Covariance Matrix Parameters

We assume that our examples have been partitioned into two sets: a training set **X**, and a holdout set $\mathbf{X}_h$. Under the assumption that the observations on our holdout set are independent and identically distributed (i.i.d.), then the optimal **Σ** is that for which the probability of correctly predicting the direction of change on all holdout examples is a maximum. We refer to this value as Σ_{MAP} (max *a posteriori*), and calculate it as

$$\begin{aligned}\Sigma_{MAP} &= \underset{\Sigma}{\arg\max} \prod_{n=1}^{|\mathbf{X}_h|} P(C_+ \mid \mathbf{x}_n, \Sigma)^{y_n} P(C_- \mid \mathbf{x}_n, \Sigma)^{1-y_n} \\ &= \underset{\Sigma}{\arg\max} \sum_{n=1}^{|\mathbf{X}_h|} y_n \ln\left(P(C_+ \mid \mathbf{x}_n, \Sigma)\right) + \left(1 - y_n\right) \ln\left(P(C_- \mid \mathbf{x}_n, \Sigma)\right)\end{aligned} \tag{6}$$

where $P(C_+ \mid \mathbf{x}_n, \Sigma)$ and $P(C_- \mid \mathbf{x}_n, \Sigma)$ are the probabilities respectively of an upward and downward movement on day *n*, and y_n is the realized direction of change on day *n*; i.e., $y_n = 1$ if $r_n > r_{n-1}$, and 0 otherwise. The problem is to find the optimal values of *a* and *k* which parameterize Σ, and in our experiments we have used a grid search optimization.

4 Empirical Results

Assuming that a return series is stationary, then a coin-flip decision procedure for predicting the direction of change would be expected to result in 50% of the predictions being correct. We would like to know whether our model can produce predictions which are statistically better than 50%. However, a problem is that many financial return series are not stationary, as evidenced by the tendency for commodity prices to rise over the long term. Thus it may be possible to achieve an accuracy significantly better than 50% by simply biasing the model to always predict up.

A better approach is to compensate for this non-stationarity, and this can be done as follows. Let x_a represent the fraction of days in an out-of-sample test period for which the *actual* movement is up, and let x_p represent the fraction of days in the test period for which the *predicted* movement is up. Therefore under a coin-flip model the expected fraction of days corresponding to a correct upward prediction is $(x_a \times x_p)$, and the expected fraction of days corresponding to a correct downward prediction is $(1{-}x_a) \times (1{-}x_p)$. Thus the expected fraction of correct predictions is

$$a_{exp} = (x_a \times x_p) + ((1{-}x_a) \times (1{-}x_p)) \,. \tag{7}$$

We wish to test whether a_{mod} (the accuracy of the predictions of our model) is significantly greater than a_{exp} (the compensated coin-flip accuracy). Thus, our null hypothesis may be expressed as follows:

Null Hypothesis: H_0: $a_{mod} \le a_{exp}$ H_1: $a_{mod} > a_{exp}$

We test this hypothesis by performing a paired one-tailed *t*-test of accuracies obtained using a collection of out-of-sample test sets from the Australian All Ordinaries (AORD) Index. Specifically, we take the period from 1 January 1992 to 31 December

2006, and divide this into 202 20-day test periods. Predictions for each of the 20-day test periods are based on a model constructed using the 250 trading days immediately preceding this test period. The number of delayed returns used for each data point was 10. For each 20-day prediction period we calculate a_{mod} and a_{exp}. We then use a paired *t*-test to determine whether the means of these values differ statistically.

We are particularly interested in observing how the significance of the results depends on the parameters *a* and *k* from Eq. 5. Table 1 shows the *t*-test *p*-values corresponding to various values for these parameters, and Table 2 shows additional information corresponding to a selection of cases from Table 1.

Table 1. *p*-values for one-sided paired *t*-test comparing a_{mod} and a_{exp}. Numbers in bold are significant at the 0.01 level. Asterisked values are the minimums for each row.

a \ *k*	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.800	1.000	5.000
0.10	0.8197	0.8462	0.8783	0.5735	0.3471	0.4935	0.2260	*0.0989	0.2169	0.3987
0.50	0.4952	0.2041	0.0212	**0.0074**	***0.0050**	**0.0065**	0.0110	**0.0016**	0.0141	0.0399
1.00	0.2430	0.1192	0.0196	0.0309	**0.0089**	***0.0038**	**0.0062**	0.0203	**0.0087**	0.0160
1.50	0.3488	0.0282	0.0313	**0.0061**	***0.0019**	**0.0016**	**0.0052**	0.0055	0.0110	0.0106
2.00	0.1036	0.0281	0.0219	**0.0006**	***0.0005**	**0.0019**	0.0109	0.0091	0.0199	0.0127
2.50	0.1009	0.1095	0.0166	**0.0087**	***0.0009**	**0.0092**	0.0186	0.0115	0.0403	0.0205
3.00	0.1012	0.0511	**0.0053**	**0.0029**	***0.0026**	**0.0071**	0.0189	0.0164	0.0672	**0.0083**
3.50	0.0913	0.0591	***0.0039**	**0.0040**	**0.0090**	0.0292	0.0486	0.0296	0.0676	**0.0094**
4.00	0.0901	0.0306	**0.0084**	***0.0056**	0.0126	0.0293	0.0841	0.0337	0.0610	**0.0059**
4.50	0.1087	0.0659	**0.0081**	***0.0037**	0.0114	0.0244	0.0479	0.0423	0.0401	**0.0073**
5.00	0.1624	0.0366	***0.0041**	**0.0058**	0.0151	0.0306	0.0502	0.0484	0.0297	**0.0060**

Table 2. Mean training accuracy, mean test accuracy, and confusion matrices corresponding to a selection of cases from Table 1. Upper/lower row of confusion matrix corresponds to upward/downward predictions; left/right column corresponds to realized upward/downward movements. Figures in parentheses are totals for rows of the confusion matrix.

	Mean Train Acc.	Mean Test Acc.	Confusion Matrix

a = 2.0
k = 0.0
	0.7830	0.5141	$\begin{bmatrix}1143 & 946\\1025 & 941\end{bmatrix}\quad\begin{matrix}(2089)\\(1966)\end{matrix}$

a = 2.0
k = 5.0
	0.5369	0.5263	$\begin{bmatrix}1005 & 757\\1163 & 1130\end{bmatrix}\quad\begin{matrix}(1762)\\(2293)\end{matrix}$

a = 0.5
k = 0.4
	0.7465	0.5245	$\begin{bmatrix}1191 & 951\\977 & 936\end{bmatrix}\quad\begin{matrix}(2142)\\(1913)\end{matrix}$

a = 2.0
k = 0.4
	0.5958	0.5343	$\begin{bmatrix}1127 & 847\\1041 & 1040\end{bmatrix}\quad\begin{matrix}(1974)\\(2081)\end{matrix}$

a = 5.0
k = 0.4
	0.5628	0.5224	$\begin{bmatrix}1009 & 777\\1159 & 1110\end{bmatrix}\quad\begin{matrix}(1786)\\(2269)\end{matrix}$

Of the values listed in Table 1, the smallest *p*-value value is 0.0005, and corresponds to parameter values *a* = 2.00 and *k* = 0.40. This means that the probability that the observed difference between a_{mod} and a_{exp} being due to chance is 0.05%, and is well below the 0.01 level commonly used to measure statistical significance. From Table 2 it can be seen that the mean test accuracy (i.e., the mean accuracy over the 202 test-periods) for this case is 0.5343, and that the mean training accuracy (i.e., the mean of the accuracy on the 202 250-day training sets) is 0.5958.

To see the effect of the use of a non-spherical covariance matrix, consider the first column of values in Table 1 ($k = 0.0$), and note that these values are much higher than the lowest p-value in the corresponding row, and yield results which are far from statistically significant (i.e., all p-values are well over 0.01). To shed further light on the role of k, consider the case $a = 2.0$, $k = 0.0$, and note that the mean training accuracy for this case is 0.7830, which is much higher than the value 0.5958 observed for $a = 2.0$, $k = 0.4$, and suggests that the model has been overfitted to the training data. Now consider right-most column of Table 1, which corresponds to $k = 5.0$. This is a large value for the scaling factor, and results in very similar p-values to what would be obtained if only a 1-dimensional delayed-return vector were used. Specifically, consider the case $a = 2.0$, $k = 5.0$. The mean training accuracy for this case is 0.5369, which is lower than that observed for case $a = 2.0$, $k = 0.4$, and suggests that underfitting is occurring. We can conclude from this that the proposed form for the kernel covariance matrix is successful in allowing recent data more influence than less recent data in the construction of the model.

To observe the effect of the parameter a, consider the last three rows of Table 2, all of which correspond to $k = 0.4$. Note that as the value of a increases, the mean accuracy on training data decreases. This can be explained by the fact that small values of a correspond to narrow kernels, which produce spiky density estimates, resulting in overfitting. Conversely, large a values result in overly smoothed densities, and thus an inability accurately model the training data.

Finally, note from Table 2 that for the cases in which a low training accuracy was achieved (e.g., $a = 2.0$, $k = 5.0$ and $a = 5.0$, $k = 0.4$), the total number of downward predicted movements is noticeably larger than the number of upward predicted movements. This can be explained by the fact that we have assumed that the priors for upward and downwards movements are equal, when in reality the priors for upward movements are higher than those for downward movements (i.e., the return series is non-stationary). When the densities are overly smoothed, the resulting posterior probability estimates are very close to the priors. If the priors for upward/downward movements were increased/decreased to reflect the fact that prices tend to rise, then the total number of upward predicted movements would increase. In fact, by pre-specifying priors it may indeed be possible to achieve out-of-sample accuracies significantly better than the value of approximately 53.4% that we have been able to achieve here. For example, if we believe that the market is displaying strong bull or bear behaviour, then we may wish to reflect this through setting the priors correspondingly.

5 Conclusions

The paper has presented a density estimation-based technique which can be used to make direction-of-change forecasts on financial time series data. A distinct advantage of the technique is that it involves very few parameters compared to discriminative models such as neural networks, and these parameters can easily be optimized using cross-validation. Also, the use of a non-spherical kernels allows recent data to have more influence than less recent data in the construction of the model, reducing the degree to which the model is sensitive to the dimensionality of the input space, thereby reducing the risk of overfitting. Results on the AORD Index show that technique is capable of yielding out-of-sample prediction accuracies which are statistically higher than those of a coin-flip procedure.

References

1. De Gooijer, J.G., Hyndman, R.J.: 25 years of time series forecasting. International. Journal of Forecasting 22, 443–473 (2006)
2. Kajitani, Y., Mcleod, A.I., Hipel, K.W.: Forecasting nonlinear time series with feed-forward neural networks: a case study of Canadian lynx data. Journal of Forecasting 24, 105–117 (2005)
3. Chung, J., Hong, Y.: Model-free evaluation of directional predictability in foreign exchange markets. Journal of Applied Econometrics 22, 855–889 (2007)
4. Christoffersen, P.F., Diebold, X., Financial, F.: asset returns, direction-of-change forecasting, and volatility dynamics. PIER Working Paper Archive 04-009, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania (2003)
5. Walczak, S.: An empirical analysis of data requirements for financial forecasting with neural networks. Journal of Management Information Systems 17, 203–222 (2001)
6. Swanson, N.R., White, H.: A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. The Review of Economics and Statistics 79, 540–550 (1997)
7. Teräsvirta, T., Medeiros, M.C., Rech, G.: Building neural network models for time series: a statistical approach. Journal of Forecasting 25, 49–75 (2006)
8. Kaastra, I., Boyd, M.S.: Designing a neural network for forecasting financial and economic time series. Neurocomputing 10, 215–236 (1996)
9. Adya, M., Collopy, F.: How effective are neural networks at forecasting and prediction? a review and evaluation. Journal of Forecasting 17, 481–495 (1998)
10. Chatfield, C.: Positive or negative? International Journal of Forecasting 11, 501–502 (1995)
11. Tkacz, G.: Neural network forecasting of canadian gdp growth. International Journal of Forecasting 17, 57–69 (2001)
12. Parzen, E.: On the estimation of a probability density function and mode. Annals of Mathematical Statistics 33, 1065–1076 (1962)
13. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley and Sons, New York (1974)
14. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)