The Performance of LVQ Based Automatic Relevance Determination Applied to Spontaneous Biosignals

Martin Golz and David Sommer

University of Applied Sciences Schmalkalden, 98574 Schmalkalden, Germany

Abstract. The issue of Automatic Relevance Determination (ARD) has attracted attention over the last decade for the sake of efficiency and accuracy of classifiers, and also to extract knowledge from discriminant functions adapted to a given data set. Based on Learning Vector Quantization (LVQ), we recently proposed an approach to ARD utilizing genetic algorithms. Another approach is the Generalized Relevance LVQ which has been shown to outperform other algorithms of the LVQ family. In the following we present a unique description of a number of LVQ algorithms and compare them concerning their classification accuracy and their efficacy. For this purpose a real world data set consisting of spontaneous EEG and EOG during overnight-driving is employed to detect so-called microsleep events. Results show that relevance learning can improve classification accuracies, but do not reach the performance of Support Vector Machines. The computational costs for the best performing classifiers are exceptionally high and exceed basic LVQ1 by a factor of 10^4.

Keywords: Automatic Relevance Determination, Learning Vector Quantization, Support Vector Machines, Electroencephalogram

1 Introduction

In many data fusion applications it was shown that combining heterogeneous sources is necessary in order to process much more information and to improve classification accuracy. In case of detecting so-called microsleep events (MSE), which are observed as attention lapses and prolonged eye lid closures during overnight driving simulations, we reported that fusion on the feature level of different sources contributes to improve the classification accuracy and the stability of the classifier [1]. Problems arise for large number of features, because all non-parametric local classification methods have fundamental problems due to the so-called "curse of dimensionality", i.e. the performance deteriorates when going to higher dimensions in the input space [2]. In this respect, simple local algorithms such as the nearest-neighbour classifier suffer more than non-local learning algorithms such as support vector machines (SVM). Note that for a given high-dimensional input vector the nearest neighbour is not much closer than other input vectors, or in other words, the ratio of the distance between the nearest and the farthest distance converges to one [2].

One way to overcome these difficulties is pruning of irrelevant features after relevance's have been gained with respect to the classification task. Models based on Bayesian statistics were proposed by MacKay [3] and Neal [4] under the terminology of Automatic Relevance Determination (ARD). ARD is also attractive as it yields simpler and more interpretable models. It is a kind of knowledge extraction in applications where the importance of features is unknown. This is also the case of MSE detection where up to now no consistent expert knowledge is available. Most known facts are related to fatigue and are not appropriate for MSE detection. Based on Learning Vector Quantization (LVQ) [5], which is a widely used and very intuitive approach to classification, three methods of ARD has been introduced in the last decade, namely distinction sensitive LVQ (DSLVQ) [6], relevance LVQ (RLVQ) [7] and generalized relevance LVQ (GRLVQ) [8]. All methods define a diagonal metric in input space which is adapted during training according to plausible heuristics. Moreover, GRLVQ benefits of a gradient dynamics on an appropriate error function. It generalizes RLVQ which is based on simple Hebbian learning and which showed worse and instable results on real world data [8].
The adaptation schemes which adjust weighting factors constitute a method for determining the intrinsic dimensionality of the data. The weighting factors can be regarded as relevance values. Dimensions with zero weight have no influence on the distances and are not relevant. Hence, the dimensions which possess the smallest relevance values are ranked as least important, i.e. they can be removed. In general, an input space dimension as small as possible is desirable for the above mentioned methods, for the sake of efficiency, accuracy, and simplicity of neural network processing.
In the same line, we proposed an adaptive metric optimization approach (in

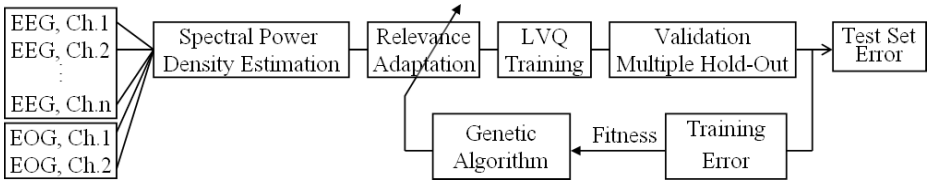

Fig. 1. Automatic Relevance Detection scheme using LVQ1 classifiers and a genetic algorithm optimizing feature weight factors in order to minimize the mean training errors [1].

the following labeled as 'GA-OLVQ1') based on the fast converging and robust OLVQ1 algorithm and a genetic algorithm (Fig. 1) [1]. Data of different sources, in our case several channels of electroencephalogram (EEG) and two channels of electrooculogram (EOG), are pre-processed and afterwards features are extracted using spectral power density estimation techniques. For each feature an individual weight factor is assigned. In the subsequent step of classification the weights are used in the distance calculation between input and prototype vectors

using weighted Euclidian metric. The classification accuracy, estimated by multiple hold-out validation of the trained networks, serves as fitness measure of a genetic algorithm. Consequently, test set errors are not used, directly and indirectly, for any step of optimization. The genetic algorithm generates populations of LVQ networks with different sets of feature weighting factors. At the end of this kind of optimization a population of well fitted LVQ networks remains. Over the ten best fitting individuals, ranked by their training errors, the weight factors are averaged. These are the final relevance values.
The purpose of this paper is to present a unique description of several algorithms of the LVQ family (section 2) and to compare their classification accuracy and the efficacy of ARD algorithms to algorithms with non-adaptive metric on a real world data set (section 4). The data set consists of spontaneous EEG and EOG during driving and is introduced in section 3. The paper is closing by some conclusions.

2 The Family of LVQ algorithms

Given a finite training set of feature vectors $x^i = (x_1, \ldots, x_n)^T$ assigned to class labels $y^i : S = \{(x^i, y^i) \subset \Re^n \times \{1, \ldots, C\} | i = 1, \ldots, m\}$ where C is the number of different classes. And given a set of randomly initialized prototype vectors w^i assigned to class labels $c^i : W = \{(w^i, c^i) \subset \Re^n \times \{1, \ldots, C\} | i = 1, \ldots, p\}$. The goal of all LVQ algorithms is to adapt the prototypes in order to yield good generalization, i.e. to make the probability of misclassification of undrawn items of a given population as small as possible.
Execute the following steps repetitively:

1. Select randomly $(x^i, y^i) \in X$ and calculate the squared distances $d_A = \|x^i - w^A\|^2_{r^o}$ and $d_B = \|x^i - w^B\|^2_{r^o}$ of x^i to the nearest and second nearest prototype vector w^A and w^B, respectively, using a weighted Euclidian metric: $\|x - w\|_{r^o} = \sqrt{\sum_{k=1}^{n} r_k^o \, |x_k - w_k|^2}$, where $r^o = (r_1, \ldots, r_n)^T$ is a normalized weight vector containing the relevance values $r_1, \ldots, r_n$.
   With respect to the class labels c^A, c^B assigned to w^A, w^B resp., and the label y^i assigned to x^i, four different cases are possible: (a) $c^A = y^i \wedge c^B \neq y^i$, (b) $c^A \neq y^i \wedge c^B = y^i$, (c) $c^A = y^i \wedge c^B = y^i$, (d) $c^A \neq y^i \wedge c^B \neq y^i$.
   Depending on these cases, a pair of factors (ν_A, ν_B) is to be chosen from table 1. They are needed in step (3) and modulate the step size and set the sense of direction of each step. For GLVQ [9] and GRLVQ the factors ν_A and ν_B are variable and are to be calculated in each iteration (Table 1).
2. Check if case (a) or case (b) is given and if x^i is in a window around the perpendicular bisector between w^A and w^B: $\min(d_A/d_B, d_B/d_A) > (\frac{1-s}{1+s})^2$, where s is the window size ($s = 0.2, \ldots, 0.3$). If these conditions are fulfilled go to step (3), otherwise do not execute update steps and go to step (1).
3. Update both prototype vectors: $\Delta w^A = \nu_A \, \eta(t) \left(x^i - w^A\right)$ and $\Delta w^B = \nu_B \, \eta(t) \left(x^i - w^B\right)$. The step size $\eta(t)$ controls the rate of convergence of the algorithm and depends on the iteration index t; $\eta(t)$ is a monotonically decreasing function.

Table 1. Iteration steps and factors (ν_A, ν_B) for different LVQ algorithms. For GLVQ and GRLVQ $\kappa_A = sgd'\left(\frac{d_A-d_B}{d_A+d_B}\right)\left[\frac{d_B}{(d_A+d_B)^2}\right]$, $\kappa_B = sgd'\left(\frac{d_A-d_B}{d_A+d_B}\right)\left[\frac{d_A}{(d_A+d_B)^2}\right]$ is to be calculated where sgd' is the first derivative of the sigmoid function. For LVQ 3, OLVQ 3 and DSLVQ the parameter κ is fixed and should be in the range $0.1, \ldots, 0.5$.

Method	Steps	Pairs of factors (ν_A , ν_B)			
		case (a)	case (b)	case (c)	case (d)
LVQ 1; OLVQ 1	(1) (3)	(1 ; 0)	(-1 ; 0)	(1 ; 0)	(-1 ; 0)
LVQ 2	(1) (2) (3)	(1 ; -1)	(0 ; 0)	(0 ; 0)	(0 ; 0)
LVQ 2.1	(1) (2) (3)	(1 ; -1)	(-1 ; 1)	(0 ; 0)	(0 ; 0)
LVQ 3; OLVQ 3	(1) (2) (3)	(1 ; -1)	(-1 ; 1)	(κ; κ)	(0 ; 0)
RLVQ	(1) (3) (4)	(1 ; 0)	(-1 ; 0)	(1 ; 0)	(-1 ; 0)
DSLVQ	(1) (2) (3) (4)	(1 ; -1)	(-1 ; 1)	(κ ; κ)	(0 ; 0)
GLVQ	(1) (2) (3)	(κ_A ; -κ_B)	(-κ_A ; κ_B)	(0 ; 0)	(0 ; 0)
GRLVQ	(1) (2) (3) (4)	(κ_A ; -κ_B)	(-κ_A ; κ_B)	(0 ; 0)	(0 ; 0)

4. Update the weight vector r^o: $\forall k = 1, \ldots, n$ do the following three steps: (i) update, (ii) threshold, (iii) normalization.
   RLVQ: (i) $r_k = r_k^o - \nu_A \eta_r \left(x_k^i - w_k^A\right)^2$, (ii) $r_k = \max(r_k, 0)$, (iii) $r^o = \frac{1}{\|r\|_2} r$
   GRLVQ: (ii) $r_k = \max(r_k, 0)$, (iii) $r^o = \frac{1}{\|r\|_2} r$
   (i) $r_k = r_k^o - \eta_r sgd'\left(\zeta \frac{d_A-d_B}{d_A+d_B}\right)\left[\zeta \frac{d_B \cdot \left(x_k^i - w_k^A\right)^2 - d_A \cdot \left(x_k^i - w_k^B\right)^2}{(d_A+d_B)^2}\right], \zeta = \begin{cases} 1, case(a) \\ -1, case(b) \\ 0, otherwise \end{cases}$
   DSLVQ:(i) $r = r^o + \eta_r\left(h^o - r^o\right)$, $h_k = \zeta \frac{\left|x_k^i - w_k^B\right| - \left|x_k^i - w_k^A\right|}{\max\left(\left|x_k^i - w_k^B\right|, \left|x_k^i - w_k^A\right|\right)}, \zeta = \begin{cases} 1, case(a) \\ -1, case(b) \\ 0, otherwise \end{cases}$
   $h^o = \frac{1}{\|h\|_1} h$, (ii) $r_k = \begin{cases} 1 \ , & r_k \geq 1 \\ 10^{-4}, & r_k \leq 10^{-4} \\ r_k \ , & otherwise \end{cases}$, (iii) $r^o = \frac{1}{\|r\|_1} r$
   Note, DSLVQ executes step(3) but not step(4) in case(c) which leads to ζ=0 [6]. R_k, x_k^i, w_k^A, w_k^B are the k-th components of r, x^i, w^A, w^B, respectively.
5. Go to step (1) until an abortion criterion is fulfilled.

The final weight vector r^o contains the relevances for each input space dimension.

3 Experimental data set

Experiments were conducted in our real car driving simulation lab. Seven EEG channels from different scalp positions (C3, C4, Cz, O1, O2, A1, A2) and two EOG-signals (vertical, horizontal) were recorded from 23 young adults during driving sessions lasting 35 minutes. These sessions were repeated every hour between 1 a.m. and 8 a.m. This way, the likelihood of the occurrence of MSE was gradually increasing due to at least 16 hours without sleep prior to the experiment.
MSE are typically characterized by driving errors, prolonged eye lid closures

or nodding-off. Towards automatic detection, two experts performed the initial MSE scoring, whereby three video cameras were utilized to record i) drivers head and upper part of the body, ii) right eye region and iii) driving scene. For further processing, only clear-cut cases, where all the experts agreed on the MSE, were taken into account. Despite providing enough test data to tune our algorithms, the human experts could not detect some of the typical attention lapses, such as the one with open eyes and stare gaze. The number of MSE varied amongst subjects and was increasing with time of day for all subjects. In all 3,573 MSE (per subject: mean number 162 ± 91, range 11-399) and 6,409 non-MSE (per subject: mean number 291 ± 89, range 45-442) were collected. Non-MSE are periods between MSE where the subject is drowsy but shows no clear or unclear MSE. This clearly highlights the need for an automated data fusion based MSE detection system, which would not only detect the MSE also recognized by human experts, but would also offer a possibility to detect the critical MSE cases which are not recognizable by human experts.
Features were extracted of 8 sec long EEG and EOG segments during MSE or non-MSE by power spectral density estimation and subsequent logarithmic scaling and averaging in frequency bands in the range from 0.5 to 35.5 Hz and a width of 1 Hz.

4 Results

In the following we want to compare between several algorithms within the LVQ family and with other classification methods applied to our real world two-class problem. The main question is addressed to classification accuracies which we estimate by computing test errors in a cross validation scheme. There is no indication that the chosen method of multiple hold-out has a remarkable estimation bias compared to the leave one-out method which is an always unbiased estimator of the true classification error [10]. In addition to the original proposed LVQ variants (LVQ1, LVQ2.1, LVQ3, OLVQ1) [5] we examine four further variants additionally executing relevance detection (DSLVQ, RLVQ, GR-LVQ, GA-OLVQ1) as mentioned above. Furthermore, we compare them also to the well-known nearest neighbour (1-NN and k-NN) algorithm, to the linear discriminant analysis (LDA), the Error Backpropagation neural network (EBP) and to the Support Vector Machine (SVM). SVM is applied using four different kernel functions because it is not known a priori which matches best for the given problem: 1) linear kernel $k(x^i, x^j) = \langle x^i, x^j \rangle$, 2) polynomial kernel: $k(x^i, x^j) = (\langle x^i, x^j \rangle + 1)^d$, 3) sigmoidal kernel: $k(x^i, x^j) = tanh(\alpha \langle x^i, x^j \rangle + \Theta)$ and 4) RBF kernel: $k(x^i, x^j) = exp(-\gamma \|x^i - x^j\|^2)$ for all x^i, $x^j \in \Re^n$.

Mean training errors and mean test errors are reported in order to quantify the ability to adapt to and to generalize the given problem (Table 1). Training errors are differing largely. Some methods are able to adapt perfectly such as 1-NN, but are not protected against overfitting. 1-NN, a typical example of a local classifier, as well as LDA, a simple global classifier, are exceeded by all LVQ variants. No important differences in the classification accuracy occurred within

Table 2. Results of multiple hold-out cross validation: mean and standard deviation of training and test errors. Different algorithms have been applied to spontaneous biosignals of the two classes "microsleep event" and "non-microsleep event". Parameters were optimized empirically. C is the regularization parameter of SVM.

Method	Parameter values	E_{TRAIN} [%]	E_{TEST} [%]
LVQ1	#neurons = 500	10.2 ± 0.2	15.7 ± 0.3
LVQ2.1	#neurons = 350	9.6 ± 0.1	15.5 ± 0.4
LVQ3	#neurons = 350	10.3 ± 0.1	15.6 ± 0.4
OLVQ1	#neurons = 500	9.3 ± 0.2	15.7 ± 0.4
RLVQ	#neurons = 500; $\eta_r = 0.01$	15.8 ± 0.4	19.5 ± 0.4
DSLVQ	#neurons = 250; $\eta_r = 0.05$	8.5 ± 0.2	15.5 ± 0.3
GLVQ	#neurons = 400	9.6 ± 0.1	15.5 ± 0.4
GRLVQ	#neurons = 350; $\eta_r = 0.01$	6.5 ± 0.2	14.3 ± 0.4
GA-OLVQ1	#generat. = 200, #popul. = 128	8.8 ± 0.2	12.9 ± 0.4
SVM linear kernel	$C = 10^{-2.75}$	15.5 ± 0.1	16.9 ± 0.2
SVM polynomial k.	$C = 10^{-2.6}$; $d = 2$	7.1 ± 0.1	14.7 ± 0.3
SVM sigmoid k.	$C = 10^{+4.4}$; $\alpha = 10^{-2.3}$; $\Theta = -1.6$	7.9 ± 0.1	12.9 ± 0.3
SVM Gaussian k.	$C = 10^{+0.31}$; $\gamma = 10^{-2.1}$	0.1 ± 0.0	10.1 ± 0.4
LDA	-	15.6 ± 0.1	17.4 ± 0.3
1-NN	-	0.0 ± 0.0	20.1 ± 0.5
k-NN	k = 11	11.6 ± 0.1	14.7 ± 0.2
EBP	#neurons = 8 (hidden layer)	12.3 ± 1.1	18.1 ± 0.7

the LVQ family, despite the RLVQ which is by 4% inferior and GRLVQ which is by 1% slightly superior. GRLVQ is outperformed by our proposed approach (GA-OLVQ1). But SVM performs still better if a Gaussian kernel function has been utilized and if the hyperparameter and the regularization parameter have been optimized.

The parameters of all applied methods have been found empirically in order to minimize test errors (Table 1). We report here only the most important parameters and their optimal values for this data set. This optimization has been done on a single training / test partition and does not influence results of other partitions. Therefore, a separate validation set is not necessary.

The computational load of the compared methods is differing largely. OLVQ1 and LVQ1 are unproblematic w.r.t. to the choice of their parameters and they have lowest computational costs, which are in the region of 10^4 iterations. This takes about 10^2 sec on a modern personal computer. This is the main reason why we used OLVQ1 in our GA-OLVQ1 approach.

RLVQ exhibits large problems when parameters are not optimal. It is very sensitive to the step size $\eta_r(t)$ (should be about 10^{-3}), otherwise RLVQ converges as fast as LVQ1. The window parameter is crucial when LVQ2, LVQ2.1, or LVQ3 is used (we have found out $s = 0.1$ to be good). They need 10 times more iterations as LVQ1 and advantageously, they perform with a lower optimal number of prototypes. DSLVQ showed the same problems and the same need of iterations as LVQ2. It is not sensitive as much as RLVQ to the choice of the step size $\eta_r(t)$. Except for the computational load, GLVQ seem to have the same properties as DSLVQ, though no metric adaptation is learned. GLVQ needs 10^6 iterations in

the average and therefore, it causes 100 times more computational costs than LVQ1. GRLVQ is in the same shape as GLVQ. The same problems as with LVQ2 are occurring and it needs about 100 times longer than LVQ1.
Our GA-OLVQ1 approach surpasses the computational costs of all other methods. It takes about 10^4 times longer than LVQ1. Therefore, we distribute the population of LVQ1 networks in a pool of 32 top modern computers and achieve a temporal consumption of about one day. The same amount of computational cost is reached by SVM because scanning for optimal values of the hyperparameter and of the regularization parameter is necessary. A single run of SVM adaptation needs about 10 times longer as for LVQ1, except when the hyperparameter value is far from the optimum. In these cases a single run of SVM can take more than 10^4 times longer as for LVQ1.
Lastly, we want to present the acquired relevance values (Fig. 2). To some ex-

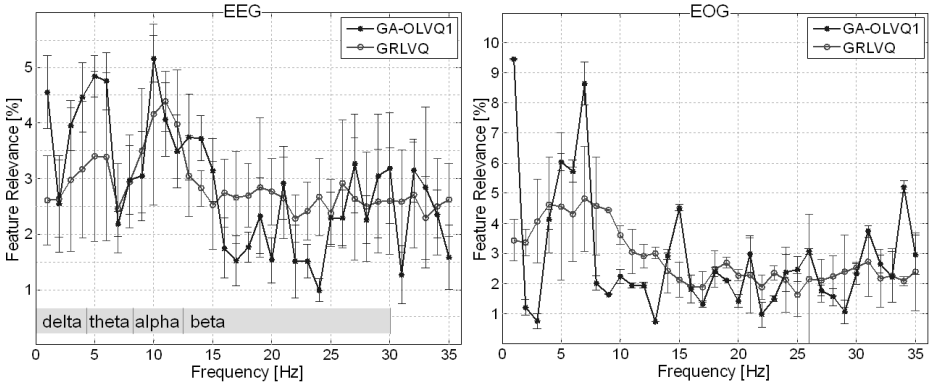

Fig. 2. Relevance values of GA-OLVQ1 and of GRLVQ for each frequency band (range: $0.5, \ldots, 35.5\,\text{Hz}, 1\,\text{Hz}$ width).

tend, they are differing between both ARD methods. The relevance values of GA-OLVQ1 show higher dynamic and mostly lower standard deviations. EEG frequencies in the region between the delta and theta band and in the alpha band, but not in the beta band are important for MSE detection. (The just now mentioned bands are common in the EEG community.) These results are in line with them in fatigue research, but the often observed downshift from alpha to high theta is in contrast to our results. In this region low relevance values were found.

5 Conclusions

We have presented an overview of some methods of the LVQ family including four approaches to automatic relevance determination. Their classification accuracy has been compared on a biomedical data set consisting of about 10^4 items.

It turned out that two of the four ARD approaches performed better than the rest of the LVQ family. Therefore, the usefulness of the underlying global metric adaptation is corroborated.
Our approach which combines all features of all the recorded EEG and EOG channels and which adapts relevance values using genetic algorithms outperformed all other LVQ methods. But best results, with test errors down to 10%, were obtained by Support Vector Machines utilizing a Gaussian kernel function and neglecting the need of metric adaptation.
Unfortunately, the computational costs of both best performing methods are exceptionally high. These costs exceed LVQ1 by a factor of 10^4.
The relevance values of the PSD features of the EEG were similar to findings of other authors in the adjacent field of fatigue research, but for a deeper understanding much more research is needed. Furthermore, there are large inter-individual differences of the EEG and EOG characteristic [1]. It would be interesting to investigate whether methods of local metric adaptation can handle this problem and therefore gaining better and more stable results than global adaptation schemes. Another future issue should be the extension to a greater variety of feature extraction methods which is also likely to improve and stabilize the MSE detection. These issues will be further steps on the long way to establish a reference measure needed for the development of video-based drowsiness warning systems.

References

1. Sommer, D., Chen, M., Golz, M., Trutschel, U., Mandic, D.: Fusion of State Space and Frequency-Domain Features for Improved Microsleep Detection. Int Conf Artificial Neural Networks (ICANN 2005); LNCS 3697, Springer, (2005) 753–759
2. Bengio, Y., Delalleau, O., Le Roux, N.: The Curse of Dimensionality for Local Kernel Machines. Techn. Rep. 1258, Université de Montréal, (2005)
3. MacKay, D. J. C.: Probable Networks and Plausible Predictions - a Review of Practical Bayesian Methods for Supervised Neural Networks. Network: Computation in Neural Systems, 6, (1995) 469–505
4. Neal, R. M.: Bayesian Learning for Neural Networks. PhD thesis, University of Toronto, Canada, LNS 118, Springer, Berlin, (1996)
5. Kohonen, T.: Self-Organizing Maps. Springer, Berlin, 3rd ed., (2001)
6. Pregenzer, M., Flotzinger, D., Pfurtscheller, G. Distinction Sensitive Learning Vector Quantization - A New Noise-Insensitive Classification Method. In Proc Int Conf Neural Networks (ICNN-94), Orlando, (1994) 2890–2894
7. Bojer, T. et al.: Relevance Determination in Learning Vector Quantization. In: M. Verleysen (ed.), Europ Symp Artif Neural Netw, D-facto public, (2001) 271–276
8. Hammer, B., Villmann, T.: Generalized Relevance Learning Vector Quantization. Neural Networks, 15 (8-9), (2002) 1059-1068
9. Sato, A.S., Yamada, K.: Generalized Learning Vector Quantization. In: Adv. Neural Inform. Process. Systems 8, MIT Press, Cambridge, (1996) 423–429
10. Sommer, D., Golz, M.: A Comparison of Validation Methods for Learning Vector Quantization and for Support Vector Machines on Two Biomedical Data Sets. in M. Spiliopoulou et al. (eds.): From Data and Information Analysis to Knowledge Engineering. Springer, (2006) 150–157