Genetically Constructed Kernels for Support Vector Machines

Stefan Lessmann[a], Robert Stahlbock[a], Sven Crone[b]

[a] University of Hamburg, Inst. of Information Systems, Von-Melle-Park 5, 20146 Hamburg, Germany
[b] Lancaster University Management School, Dept. of Management Science, Lancaster, LA1 4YX, United Kingdom

Abstract

Data mining for customer relationship management involves the task of binary classification, e.g. to distinguish between customers who are likely to respond to direct mail and those who are not. The support vector machine (SVM) is a powerful learning technique for this kind of problem. To obtain good classification results the selection of an appropriate kernel function is crucial for SVM. Recently, the evolutionary construction of kernels by means of meta-heuristics has been proposed to automate model selection. In this paper we consider genetic algorithms (GA) to generate SVM kernels in a data driven manner and investigate the potential of such hybrid algorithms with regard to classification accuracy, generalisation ability of the resulting classifier and computational efficiency. We contribute to the literature by: (1) extending current approaches for evolutionary constructed kernels; (2) investigating their adequacy in a real world business scenario; (3) considering runtime issues together with measures of classification effectiveness in a mutual framework.

1 Introduction

The support of managerial decision making in marketing applications is a common task for corporate data mining with classification playing a key role in this context [2]. The SVM [9] is a reliable classifier that has been successfully applied

to marketing related decision problems, e.g. [1; 10]. Like other learning algorithms such as neural networks, the SVM algorithm offers some degrees of freedoms that have to be determined within the data mining process. The selection of suitable parameters is crucial for effective classification. Therefore, we propose a data driven heuristic to determine the SVM parameters without manual intervention.

The remainder of this paper is organised as follows: Following a brief introduction to SVM theory we present our combination of GA and SVM (GA-SVM) in Section 3. The potential of GA-SVM is evaluated in a real world scenario of direct marketing in Section 4. Conclusions are given in Section 5.

2 Support Vector Machines

The SVM is a supervised learning machine to solve linear and non-linear classification problems. Given a training set $S = \{\boldsymbol{x}_i; y_i\}_{i=1}^{m}$ where $\boldsymbol{x}_i$ is a n-dimensional real vector and $y_i \in \{-1,+1\}$ its corresponding class label, the task of classification is to learn a mapping $\boldsymbol{x}_i \mapsto y_i$ from *S*, that allows the classification of new examples with unknown class membership.

The SVM is a linear classifier of the form

$$y(\boldsymbol{x}) = sgn(\boldsymbol{w} \cdot \boldsymbol{x} + b), \tag{1}$$

which strives to maximise the margin of separation between the two classes [9]. The parameters $\boldsymbol{w}$ and b realising such a maximal margin hyperplane can be found by solving a quadratic optimisation problem with inequality constraints; e.g. [3].

In order to derive more general, non-linear decision surfaces SVMs implement the idea to map the input data into a high-dimensional feature space via an a priori chosen non-linear mapping function. Due to the fact, that the SVM optimisation problem contains the input patterns only as dot products, such a mapping can be accomplished implicitly by introducing a kernel function [3; 9]

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j). \tag{2}$$

Beside the selection of an appropriate kernel and its corresponding kernel parameters, see Section 3, the SVM classifier offers one additional regularisation parameter *C* which controls the trade off between maximising the margin of separation and classifying the training set without error.

3 Genetic algorithms for SVM model selection

The classification performance of SVM depends heavily on the choice of a suitable kernel function and an adequate setting of the regularisation parameter *C*.

Consequently, we develop a data driven approach to determine the kernel *K* and its corresponding kernel parameters together with *C* by means of GA. Using the five basic kernels of Table 1, we construct a combined kernel function as

$$K_{poly}^{1} \otimes K_{rad}^{\alpha} \otimes K_{sig}^{\beta} \otimes K_{imq}^{\gamma} \otimes K_{anova}^{1} \quad , \tag{3}$$

with $\otimes \in \{+;\cdot\}$, where we exploit the fact that if K_1 and K_2 are kernels, $K_1 + K_2$ and $K_1 \cdot K_2$ are valid kernels as well [3].

Table 1. Basic SVM kernel functions

Polynomial kernel	$K_{poly}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(a(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + b\right)^c$
Radial kernel	$K_{rad}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-a\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right)$
Sigmoidal kernel	$K_{sig}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(a(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + b\right)$
Inverse multi-quadratic kernel	$K_{imq}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1/\sqrt{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 + b^2}$
Anova kernel	$K_{anova}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_j \exp\left(-a\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right)^2\right)^c$

To encode (3) into a structure suitable for GA based optimisation we use five integer genes for the kernel exponents in (3), four binary genes for the kernel combination operator $\otimes$ and sixteen real-valued genes for the specific kernel parameters (three per kernel) as well as the regularisation parameter *C*. The complete structure is given in Fig. 1. This coding is inspired by [7] and extends their approach to five kernels and the inclusion of C into the GA based optimisation.

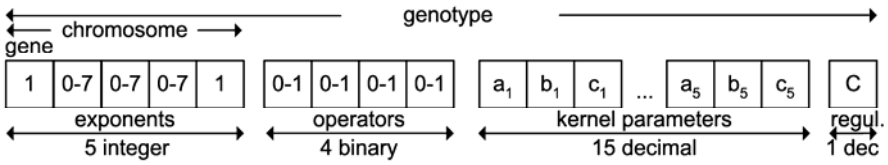

Fig. 1. Structure of the genotype for SVM kernel construction

The GA is implemented in accordance with [8] and utilises a uniform crossover for the five kernel exponent genes. That is, all genes between two random points within this string are interchanged between two genotypes representing parents for the resulting two new genotypes. The mutation operator is implemented as a simple bit swap for the four kernel combination genes and a random increment or decrement for all integer and real value genes. Crossover and mutation probabilities have been determined through pre-tests to 0.7 and 0.3 respectively.

4 Empirical evaluation

4.1 Experimental setup

The simulation experiment aims at comparing genetically constructed SVM with conventional ones to assess capabilities of GA to support SVM model selection.

We consider the case of repeat purchase modelling in a direct marketing setting, see e.g. [1; 10], using real world data from a German publishing house. The data set consists of 300,000 customer records that have been selected for a past mailing campaign to cross-sell an additional magazine subscription to customers that have subscribed to at least one periodical. Each customer is described by a 28-dimensional vector of 9 numerical and 19 categorical attributes describing transactional and demographic customer properties. The number of subscriptions sold in this campaign is given with 4,019, leading to a response rate of 1.35% which is deemed to be representative for the application domain. An additional target variable indicates the class membership of each customer (class 1 for subscribers and class -1 for non subscribers) facilitating the application of supervised learning algorithms to model a relationship between customer attributes and likelihood of responding to direct mail.

Classifiers are evaluated applying a hold-out method of three disjoint datasets to control over-fitting and for out-of-sample evaluation. While training data is used for learning, i.e. determining the decision variables $\boldsymbol{w}$ and b, see (1), a validation set is used to steer the GA. That is, a classifier's performance on the validation set represents its fitness and is used to select items for the mating pool within the GA [4]. The trained and selected classifiers are finally tested on an unknown hold-out set to evaluate their generalisation ability on unknown data.

In order to assure computational feasibility and with regard to the vast imbalance between class 1 and class -1 membership within our data set, we apply an undersampling approach [11] to obtain a training and validation data set of 4,144 and 2,070 records respectively with equal class distributions. The test set consists of 65,000 records containing 912 class 1 customers, reflecting the original unequal distribution of the target variable.

4.2 Experimental results

In order to deliver good results GA usually require a large population size that ensures sufficient variability within the elements in the gene pool [8]. For GA-SVM we select a population size of 50 and monitor the progress in classification quality for 15 generations. Thus, 750 individual SVMs with genetic kernel are constructed on the training set, assessed on the validation set and finally evaluated on the test set. Since the skewed class distribution of the target variable prohibits the application of standard performance metrics of classification accuracy [11], we used the G-metric instead [6]. Striving to maximise the class individual accuracies while keeping them balanced the G-metric is calculated as the geometric mean between

class individual accuracies. Consequently, higher values indicate improved predictive accuracy.

Results at the generation level are given in Table 2 where each value is calculated on the basis of the 50 individual GA-SVM classifiers within a generation.

Table 2. Results of GA-SVM at the generation level over 15 generations

	Mean runtime per SVM [min]		SVM performance by means of G-metric on training set		validation set		test set	
Generation	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.
0	91.3	53.4	0.596	0.306	0.544	0.277	0.444	0.225
1	71.2	37.0	0.731	0.158	0.661	0.145	0.534	0.111
2	78.1	38.8	0.687	0.236	0.633	0.215	0.496	0.168
3	77.8	27.9	0.754	0.158	0.685	0.142	0.528	0.110
4	79.6	31.1	0.736	0.192	0.668	0.172	0.516	0.132
5	76.0	27.8	0.759	0.158	0.684	0.142	0.527	0.110
6	68.8	16.6	0.786	0.025	0.713	0.019	0.549	0.013
7	77.3	31.9	0.785	0.030	0.714	0.015	0.547	0.012
8	67.8	22.8	0.775	0.114	0.703	0.102	0.537	0.078
9	65.1	21.7	0.768	0.115	0.696	0.105	0.539	0.079
10	67.8	25.0	0.784	0.034	0.711	0.027	0.552	0.012
11	64.2	11.2	0.795	0.008	0.721	0.012	0.551	0.009
12	62.2	12.5	0.796	0.008	0.720	0.015	0.552	0.009
13	59.6	12.5	0.791	0.014	0.716	0.019	0.553	0.010
14	59.4	12.6	0.789	0.014	0.720	0.015	0.553	0.008

Our results show a generally increasing average performance from generation to generation over all data sets. However, vast improvements are obtained only when moving from generation 0 to 1, indicating that a saturation level is reached early in the evolutionary process. In fact, while a oneway analysis of variance confirmed a highly significantly difference in mean performance over all data sets at the 0.001 level, a Tukey post hoc test revealed that only the generations 0 and 2 differ from the remaining ones significantly at the 0.01 level.

The decrease in standard deviation is more explicit and illustrates a higher similarity within the gene pool. Interestingly, the average runtimes decrease tremendously, meaning that the high quality kernels of later generations are also computationally more efficient. The best kernel was found in generation 14 with a test set G-value of 0.585 incorporating all base kernels but the anova kernel.

To compare our approach with standard SVM we calculate solutions for the radial and polynomial SVM classifier, conducting an extensive grid search [5] in the range $\log(C) = \{-4; 4\}$ and $\log(a) = \{-4; 4\}$ with a step size of one for the radial kernel and $\log(C) = \{-2; 3\}$, $\log(a) = \{-2; -1\}$, $b = \{0; 1\}$, $c = \{2; 7\}$ for the polynomial kernel to obtain and average G-value of $G_{radial} = (0.70; 0.58; 0.53)$ and $G_{polynomial} = (0.71; 0.65; 0.54)$ on training, validation and test sets. As expected, the higher flexibility of the combined kernel in GA-SVM allows a purer separation of the training set. Regarding generalisation, GA-SVM consistently outperforms classical SVM in later generations, providing superior results on the validation set from generation 3 and on the test set from generation 10 onwards.

5 Conclusions

We investigated the potential of SVMs with GA-optimised kernel functions in a real world scenario of corporate decision making in marketing. Solving more than 750 evolutionary constructed SVMs, the GA proved to be a promising tool for kernel construction, enhancing the predictive power of the resulting classifier. However, the vastly increased computational cost might be the main obstacle for practical applications. Most radial SVMs needed less than a minute to construct a solution and the runtime of polynomial SVMs ranged from 12 to 60 minutes. In contrast, we observed average GA-SVM runtimes of 60 to 90 minutes.

Since the task of model selection shifts from setting SVM parameters to determining the parameters of the utilised search heuristic, the proposed GA is a promising candidate for SVM tuning, offering only four degrees of freedom on its own (crossover and mutation probabilities, population size, termination criterion e.g. number of generations).

Further research involves the application of GA-SVM to other data sets as well as a detailed analysis and comparison of the constructed kernels per generation.

References

[1] Baesens B, Viaene S, Van den Poel D, Vanthienen J, Dedene G (2002) Bayesian neural network learning for repeat purchase modelling in direct marketing. European Journal of Operational Research 138(1):191-211

[2] Berry MJA, Linoff G (2004) Data mining techniques: for marketing, sales and customer relationship management, 2. edn. Wiley, New York

[3] Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

[4] Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading

[5] Keerthi SS, Lin C-J (2003) Asymptotic Behaviours of Support Vector Machines with Gaussian Kernel. Neural Computation 15(7):1667-1689

[6] Kubat M, Holte RC, Matwin S (1998) Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning 30(2-3):195-215

[7] Nguyen H-N, Ohn S-Y, Choi W-J (2004) Combined Kernel Function for Support Vector Machine and Learning Method Based on Evolutionary Algorithm. In: Pal NR, Kasabov N, Mudi RK (eds) Proc. of the 11th Intern. Conf. on Neural Information Processing, Calcutta, India, pp 1273-1278

[8] Stahlbock R (2002) Evolutionäre Entwicklung künstlicher neuronaler Netze zur Lösung betriebswirtschaftlicher Klassifikationsprobleme. WiKu, Berlin

[9] Vapnik VN (1995) The Nature of Statistical Learning Theory. Springer, New York

[10] Viaene S, Baesens B, Van Gestel T, Suykens JAK, Van den Poel D, Vanthienen J, De Moor B, Dedene G (2001) Knowledge discovery in a direct marketing case using least squares support vector machines. International Journal of Intelligent Systems 16(9):1023-1036

[11] Weiss GM (2004) Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6(1):7-19