Improving the Classification Performance of Boolean Kernels by Applying Occam's Razor

Yang Zhang, Zhanhuai Li, Muning Kang, Jianfeng Yan
Dept. Computer Science & Engineering,
Northwestern Polytechnical University, P. R. China
{zhangy, lzh, mnkang}@co-think.com
jfyan@mail.nwpu.edu.cn

Abstract

Support vector machines could be used to create classifiers by learning Boolean functions in high dimensional feature space when using Boolean kernels. However, currently the classification performance of Boolean kernels is poor. In this paper we present three Boolean kernels, and we also reveal the inductive bias embodied in these kernels. In order to improve the classification performance of the Boolean kernels, we present an approach to apply the Boolean kernels with the inductive bias of Occam's razor. Experiment result shows that the classification performance of Boolean kernels could be improved by our approach.

1 Introduction

Classification, which is a primary data mining task, is to categorize the sample data into one of the several pre-defined categories. Non-linear Support Vector Machine (SVM) could be used to create binary classifiers. The non-linear SVM maps the vectors in the input space to vectors in a high dimensional space, which is named feature space, by a non-linear mapping, and constructs an optimal separating hyperplane, which could separate positive sample and negative sample from each other in the feature space.

If the sample data for the classification task only have discrete attributes, then each sample could be represented by a Boolean vector, and each dimension of the input space could be represented by a Boolean literal. Suppose the input space is $\{0,1\}^n$, if there exists a non-linear mapping f, which can map the vectors in the input space to vectors in a feature space, so that each dimension in the feature space could be looked as the conjunctive of positive Boolean literals or negative Boolean literals in the input space, then the SVM could construct an optimal separating hyperplane for classification in the feature space, and the classifier could be looked as the weighted linear sum of these conjunctions of Boolean literals. So, SVM could be used to learn Boolean functions for classification [1]. Taking $\{0,1\}^n$ as the input space, a Boolean kernel is a function that can calculate the inner product of two vectors in the feature space even without the prior knowledge of f.

However, the classification performance of a SVM using Boolean kernels for learning Boolean functions degrades rapidly when the dimension of input space becomes large. This is because the size of hypothesis space grows exponentially as the dimension of input space grows, and the learning machine overfits to training data with high probability [2].

Inductive bias is some form of prior assumptions. The inductive bias that is widely used in decision tree algorithm is that shorter trees are preferred over larger trees. This inductive bias is also known as *Occam's razor* [3,4]. This may suggest that the classification performance of the Boolean kernels could be improved by emphasizing the contribution to the classifier made by short conjunctions of Boolean literals, and by reducing the contribution made by long conjunctions.

In this paper, we present three Boolean kernels that can be used to create classifiers by learning Boolean functions, and reveal the inductive bias embodied in these kernels. We also present an approach to improve the classification performance of Boolean kernels by applying the Boolean kernels with the inductive bias of *Occam's razor*. Experiment results show that the classification performance of Boolean kernels could be improved by this approach.

The remainder of this paper is organized as follows: Section 2 reviews the related works, Section 3 presents three Boolean kernels for classification by learning Boolean functions, and an approach to improve the classification performance of Boolean kernels, Section 4 presents our experimental results, Section 5 makes conclusion and suggests our further work.

2 Related Works

[5] and [6] are tutorials for SVM and kernel machines. [7,8] are devoted to the study of learning Boolean functions, with [7] focusing on the learnable or non-learnable of Boolean functions, and [8] focusing on the learning algorithms. Studies about learning Boolean functions by SVM could be found in [1,2,9,10], with [1,2] focusing on classification task. In [1,9,10], several Boolean kernels were presented. In the knowledge of us, [2] is the only study that focuses on the approach to improve the classification performance of Boolean kernels. In [2], Sadohara presented the BKC algorithm, which classifies by conjunctions of no more than k Boolean literals, and the parameter *k* is learned automatically from training dataset.

The difference between BKC algorithm and our approach is that BKC algorithm only uses conjunctions with length up to *k*, and that BKC algorithm treats the contribution to classification made by these conjunctions equally, while in our approach we use all conjunctions for classification, but we emphasize the contribution made by short conjunctions and reduce the contribution made by long conjunctions. A detailed comparison of BKC and our approach is postponed to Section 4.

3 Boolean Kernels for Classification

In this paper, we focus on classification tasks when sample data only have discrete attributes. Numerical attributes could be discretized into discrete attributes. For a discrete attribute A_i, if there are $|A_i|$ possible values for this attribute, then we can use $|A_i|$ Boolean literals to represent this attribute, with each Boolean literal representing the occurrence or non-occurrence of the corresponding attribute value. So, a sample dataset with d samples could be represented as $\{X_i, y_i\}$, i=1, 2,…,d, with $X_i \in \{0,1\}^n$, representing a sample data, and $y_i \in \{-1,+1\}$, representing the class type of this sample. In the following discussion, we take each dimension in the input space as a Boolean literal, which takes value from {0,1}.

By using Boolean kernels, SVM constructs a classifier $f(X) = \langle W, f(X) \rangle + b$ in the feature space. (Here, W is a vector in the feature space and $b \in R$.) For the Boolean kernels presented in [1,10], the non-linear mapping f makes each dimension in the feature space correspond to the conjunction of several positive Boolean literals or negative Boolean literals, and the kernels treat the contribution to classification made by these conjunctions equally. When the dimension of input space becomes large, the dimension of feature space expands exponentially. The huge amount of features in the feature space disturbs SVM, making its classification performance decrease drastically.

The theory of *Occam's razor* works successfully when building decision tree classifiers, which suggests that in the feature space SVM should emphasizes the contribution to classification made by short conjunctions, while reducing the contribution made by long conjunctions.

Here we present three Boolean kernels that can learn Boolean functions for classification, and present an approach of applying the Boolean kernels with the inductive bias of Occam's razor, so as to improve their classification performance.

3.1 Monotone DNF Kernel

Proposition1.Suppose $U \in \{0,1\}^n, V \in \{0,1\}^n, s > 0$, then $K_{MDNF}(U,V) = -1 + \prod_{i=1}^{n} (\sigma U_i V_i + 1)$ is a Boolean kernel.

Proof. Suppose $X \in \{0,1\}^n$, let's consider the function $f(X) = -1 + \prod_{i=1}^{n} (\sqrt{s}\, X_i + 1)$. There are 2^n-1 terms in the expansion of $f(X)$, and each term could be looked as the conjunction of several Boolean literals in the input space. So, $f(X)$ could be looked as the weighted linear sum of all possible conjunctions of positive Boolean literals. Let f_{MDNF} be the non-linear mapping from $\{0,1\}^n$ to R^{2^n-1}, with each dimension in the feature space being $\prod_{i=1}^{n} (\sqrt{s}\, X_i)^{s_i}$ (here, $s_i \in \{0,1\}$ $i=1,2..,n$, $\sum_{i=1}^{n} s_i \neq 0$). Then, each dimension in the feature space corresponds to a term in the expansion of $f(X)$. So, the inner product in the feature space satisfies $\langle f_{MDNF}(U), f_{MDNF}(V) \rangle = K_{MDNF}(U,V)$. Therefore, $K_{MDNF}(U,V)$ is a Boolean kernel. ■

Suppose $Z \in R^n, W \in R^n$, let consider the function $f(Z,W) = \langle Z, W \rangle$. We can get $\frac{\partial f}{\partial Z_i} = W_i$, which means that for a certain value of Z_i, the bigger $|W_i|$ is, the more contribution Z_i could make to $f(Z,W)$. This conclusion could be used for feature selection for linear

SVM, and it works successfully when it is applied to text classification [11]. Here, we use this conclusion for feature selection in the feature space.

Suppose $X \in \{0,1\}^n, W \in R^{2^n-1}, b \in R$, let's consider the function $f(X,W) = \langle f_{MDNF}(X), W \rangle + b$. In the expansion of $f(X,W)$, the coefficient for the conjunction with s (s=1,2,…,n) Boolean literals, $X_{i_1} X_{i_2} \ldots X_{i_s}$, is $s^{\frac{s}{2}} W_j$. (Here, i_l=1,2,…,n , l=1,2,…,s). When $\sigma > 1$, the bigger s is, the more contribution $X_{i_1} X_{i_2} \ldots X_{i_s}$ could make to $f(X,W)$. Especially, we can get $s^s > s^{s-1}$, which means that compared with the contribution to $f(X,W)$ made by conjunctions with length s-1, the contribution made by conjunctions with length s is emphasized. So, when $s > 1$, the inductive bias introduced by s is to emphasize the contribution to classification made by long conjunctions, and the bigger s is, the stronger this emphasis is. Similarly, we can draw conclusion that when $0 < s < 1$, the inductive bias introduced by s is to emphasize the contribution to classification made by short conjunctions, and the less s is, the stronger this emphasis is. When $s = 1$, we get the Monotone Disjunctive Normal Form (MDNF) kernel as defined in [1], which treats the contribution made by conjunctions with different length equally.

In the training phase, SVM will emphasize the contribution to classification made by short conjunctions if we set $0 < \sigma < 1$, so as to apply the inductive bias of *Occam's razor* in $K_{MDNF}(U,V)$.

The time complexity of $K_{MDNF}(U,V)$ is O(n).

3.2 Polynomial Kernel

Proposition 2. Suppose $U \in \{0,1\}^n, V \in \{0,1\}^n, \sigma > 0$, $p \in N$, then $K_{POLY}(U,V) = (\sigma \langle U, V \rangle + 1)^p - 1$ is a Boolean kernel.

Proof. Suppose $X \in \{0,1\}^n$, let's consider the function $f(X) = (\sqrt{s}\langle X, 1 \rangle + 1)^p - 1$. Each term in the expansion of $f(X)$ is $\sqrt{\frac{p!}{s_1! s_2! .. s_n! s_{n+1}!}} (\sqrt{s})^{s_1+s_2+\ldots+s_n} X_1^{s_1} X_2^{s_2} \ldots X_n^{s_n}$, here, $s_i \in \{0, N\}$, $i = 1,2,..n,n+1$, $\sum_{i=1}^{n} s_i > 0$, $\sum_{i=1}^{n+1} s_i = p$.

Let f_{POLY} be the mapping from $\{0,1\}^n$ to a high dimensional feature space, with each dimension in the feature space being $\sqrt{\frac{p!}{s_1! s_2! .. s_n! s_{n+1}!}} (\sqrt{s})^{s_1+s_2+\ldots+s_n} X_1^{s_1} X_2^{s_2} \ldots X_n^{s_n}$, which corresponds to a term in the expansion of $f(X)$. So, the inner product in the feature space satisfies $\langle f_{POLY}(U), f_{POLY}(V) \rangle = K_{POLY}(U,V)$. Therefore, $K_{POLY}(U,V)$ is a Boolean Kernel. ■

Suppose W is a vector in this feature space, $X \in \{0,1\}^n$, let's consider the function $f(X,W) = \langle f_{POLY}(X), W \rangle$. Each term in the expansion of $f(X,W)$ is in the form $\sqrt{\frac{p!}{s_1! s_2! .. s_n! s_{n+1}!}} (\sqrt{s})^{s_1+s_2+\ldots+s_n} X_1^{s_1} X_2^{s_2} \ldots X_n^{s_n} W_j$.

Here, we can take $X_1^{s_1} X_2^{s_2} \ldots X_n^{s_n}$ as $\underbrace{X_1 \wedge X_1 \ldots \wedge X_1}_{S_1} \wedge \underbrace{X_2 \wedge X_2 \ldots \wedge X_2}_{S_2} \wedge \cdots \wedge \underbrace{X_n \wedge X_n \ldots \wedge X_n}_{S_n}$, a conjunction of $\sum_{i=1}^{n} s_i$ Boolean literals. Similar to the deduction in 3.1, we can draw conclusion that when $0 < s < 1$, the inductive bias introduced by s is to emphasize the contribution to classification made by short conjunctions, and the less s is, the stronger this emphasis is.

The time complexity of $K_{POLY}(U,V)$ is O(n).

The difference between $K_{MDNF}(U,V)$ and $K_{POLY}(U,V)$ is that $K_{MDNF}(U,V)$ uses conjunctions of no more than n Boolean literals for classification, while $K_{POLY}(U,V)$ use conjunctions of no more than p Boolean literals. The similarity of them is that both of these two kernels only use positive Boolean literals for classification. The following kernel use both positive Boolean literals and negative Boolean literals for classification.

3.3 DNF Kernel

Proposition 3. Suppose $U \in \{0,1\}^n, V \in \{0,1\}^n, s > 0$, then $K_{DNF}(U,V) = -1 + \prod_{i=1}^{n}(sU_iV_i + s(1-U_i)(1-V_i) + 1)$ is a Boolean kernel.

Proof. Suppose $X \in \{0,1\}^n$, let's write $\overline{X_i}$ for $1 - X_i$, and consider the function $f(X) = -1 + \prod_{i=1}^{n}(\sqrt{s}\,X_i + \sqrt{s}\,\overline{X_i} + 1)$. There are 3^n-1 terms in the expansion of $f(X)$, and each term could be looked as conjunctions of several positive Boolean literals or negative Boolean literals in the input space. $f(X)$ could be looked as the weighted linear sum of all possible conjunctions of positive Boolean literals or negative Boolean literals. Let f_{DNF} be a mapping from $\{0,1\}^n$ to R^{3^n-1}, with each dimension in the feature space being $\prod_{i=1}^{n}(\sqrt{s}\,X_i)^{s_i}(\sqrt{s}\,\overline{X_i})^{1-s_i}$, (here, $s_i \in \{0,1\}$ $i = 1,2,...,n$, $\sum_{i=1}^{n} s_i \neq 0$), which corresponds to a term in the expansion of $f(X)$. So, the inner product in the feature space satisfies $\langle f_{DNF}(U), f_{DNF}(V) \rangle = K_{DNF}(U,V)$. Therefore, $K_{DNF}(U,V)$ is a Boolean kernel. ■

From a similar discussion as in section 3.1, we can draw conclusion that when $0 < s < 1$, the inductive bias introduced by s is to emphasize the contribution to classification made by short conjunctions, and the less σ is, the stronger this emphasis is. When $s = 1$, we get the Disjunctive Normal Form (DNF) kernel as defined in [1].

The time complexity of $K_{DNF}(U,V)$ is O(n).

4 Experiment Result

We chose a text dataset, Reuters21578 [1], for experiment because text classification tasks always have a high dimensional input space. Text collections for experiments are usually split into two parts: a training dataset for building the classifier and a testing dataset for testing the performance of the classifier. There are many splits of the Reuters collection, and we chose to use the *ModApte* version, which is widely used by researchers [12]. Many researchers only use the ten most populated document classes in this dataset for experiment. We prepare our dataset following this way. The following detailed experiment steps are repeated for each of the ten document classes.

1. Preprocessing data, including extracting text document from the dataset, stop-word removing [2] and stemming[3].
2. Selecting the top 100 features with the highest information gain (IG), which is a feature selection algorithm widely used in text classification [12].
3. Representing text document by a binary vector with these 100 features.
4. Building SVM classifiers from training dataset, and classifying testing dataset by these classifiers.

In the experiment, we use SVM^{light} software package[4] as the SVM learner and classifier by plugging our kernels into this package.

Recall and precision can be used for measuring performance of text classifiers. F_1 measure is widely used by researchers because it can measure recall and precision at the same time. Here, we report our experiment result in micro-average F_1 and accuracy. Suppose in the test dataset, the number of positive samples that has been classified correctly is *TP*; the number of negative samples that has been classified correctly is *TN*; the number of negative samples that has been classified into positive class is *FP*; the number of positive samples that has been classified into negative class is *FN*; then precision=*TP*/(*TP*+*FP*); recall=*TP*/(*TP*+*FN*); F_1=2*pecision*recall/(precision+recall); accuracy=(*TP*+*TN*)/(*TP*+*FP*+*TN*+*FN*). Micro-average F_1 is calculated from *TP*, *TN*, *FP* and *FN* that are got after the experiment had been repeated on all of the 10 document classes [12].

Figure 1, 2 gives the experiment result for $K_{MDNF}(U,V)$ and $K_{DNF}(U,V)$ respectively. In these figures, the horizontal axis represents the parameter s; the vertical axis represents the experiment results. *Micro F1* represents the result measured in micro-average F_1, and *Accuracy* represents the result measured in accuracy.

[1] URL http://www.daviddlewis.com/resources/testcollections/reuters21578/

[2] URL http://www.tartarus.org/~martin/PorterStemmer/

[3] URL ftp://sunsite.dcc.uchile.cl/pub/users/rbaeza/irbook/stopper/

[4] URL http://svmlight.joachims.org/

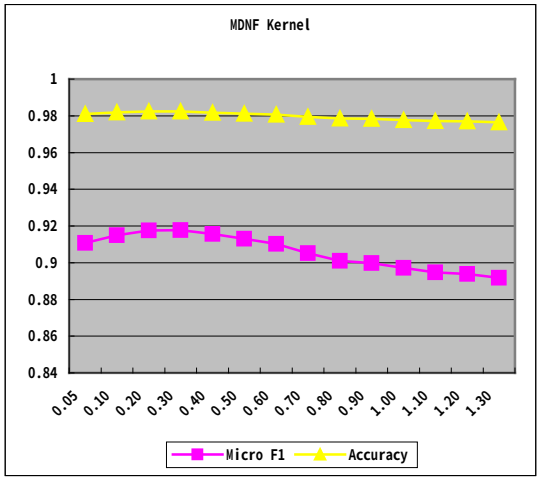

Figure 1: Experiment Result for $K_{MDNF}(U,V)$.

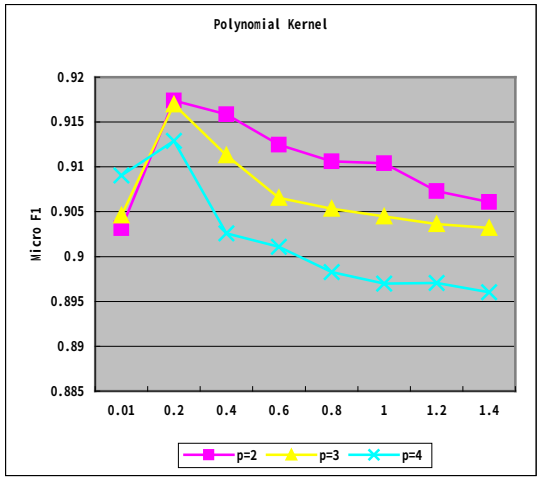

Figure 3: Experiment Result in Micro-average F_1 for $K_{POLY}(U,V)$.

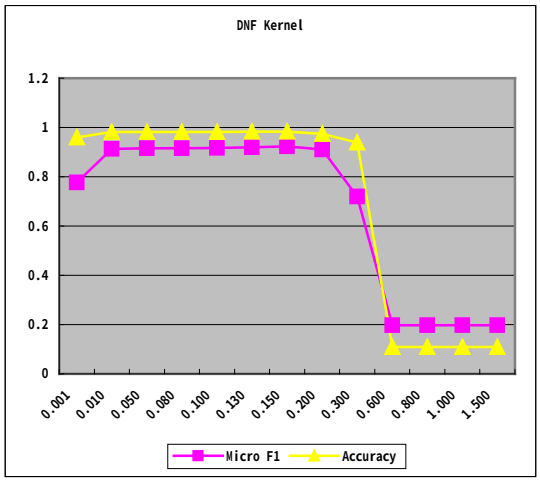

Figure 2: Experiment Result for $K_{DNF}(U,V)$.

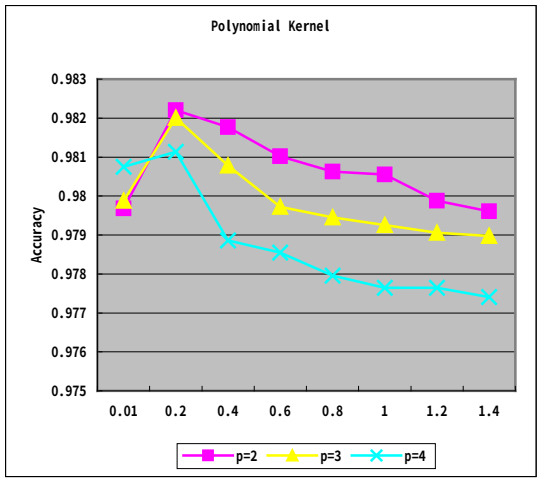

Figure 4: Experiment Result in Accuracy for $K_{POLY}(U,V)$.

	BKC-MDNF			$K_{MDNF}(U,V)$			BKC-DNF			$K_{DNF}(U,V)$		
	T=10	T=20	T=30	σ =0.2	σ =0.3	σ =0.4	T=10	T=20	T=30	σ =0.13	σ =0.14	σ =0.15
Micro F1	0.9114	**0.9208**	0.9148	0.9176	**0.9177**	0.9156	0.9138	**0.9147**	**0.9147**	0.9198	0.9214	**0.9230**
Accuracy	98.08 %	**98.25 %**	98.12 %	98.23 %	**98.23 %**	98.18 %	98.13 %	**98.15 %**	**98.15 %**	98.25 %	98.29 %	**98.32 %**

Table 1: The Best Classification Result for BKC, $K_{MDNF}(U,V)$ and $K_{DNF}(U,V)$.

Figure 3, 4 gives the experiment result for $K_{POLY}(U,V)$ measured in micro-average F_1 and accuracy respectively. In these figures, the horizontal axis represents the parameter s ; the vertical axis represents the experiment results. *p=2* represents the experiment result when the parameter p in $K_{POLY}(U,V)$ is set to 2. Others are similar.

From these figures, it is obviously that for all of the three Boolean kernels discussed in this paper, the classification performance when $s < 1$ is better than the performance when $s > 1$. The reason is that when $s < 1$ the inductive bias introduced by s is to emphasize the contribution to classification made by short conjunctions. In other words, the classification

performance is improved by applying the inductive bias of *Occam's razor* in the Boolean kernels. When s is too small, the SVM puts too much emphasis on the contribution made by short conjunctions, which makes the classification performance degrades rapidly.

In [2], Sadohara proposed the BKC algorithm for improving the classification performance of Boolean kernels. Table 1 gives the best experiment result by BKC algorithm, $K_{MDNF}(U,V)$ and $K_{DNF}(U,V)$. Here, *BKC-MDNF* represents the classification result for BKC algorithm that creates classifiers using conjunctions of only positive Boolean literals; *BKC-DNF* represents the classification result for BKC algorithm that creates classifiers using conjunctions of both positive and negative Boolean literals; $K_{MDNF}(U,V)$ and $K_{DNF}(U,V)$ represents the result for $K_{MDNF}(U,V)$ and $K_{DNF}(U,V)$, respectively. *T=20* represent the result when BKC algorithm use conjunctions of no more than 20 Boolean literals for classification. Others are similar. The results measured in micro-average F_1 and accuracy are both presented. We can see that for classifiers that use only positive Boolean literals for classification, the BKC algorithm has a better classification performance than the approach proposed in this paper; and for classifiers that use both positive and negative Boolean literals for classification, the approach proposed in this paper has a better performance than BKC algorithm.

The time complexity of Boolean kernels used for BKC algorithm is O(kn) [2]. Here, the parameter k is learned from training dataset by BKC algorithm automatically. The time complexity of Boolean kernels proposed in this paper is O(n).

5 Conclusion and Future work

In this paper, we present three Boolean kernels that can learn Boolean function for classification in the high dimensional feature space. We also represent an approach to improve the classification performance of Boolean kernels by applying the inductive bias of *Occam's razor*. Experiment results show that the performance of Boolean kernels can be improved by our approach.

The classification performance of Boolean kernels proposed in this paper is very competitive as compared with BKC algorithm, which also uses Boolean kernels for classification. The time complexity of the Boolean kernels proposed in this paper is O(n), while the time complexity of Boolean kernels used for BKC algorithm is O(kn).

The parameter s plays an important role in the classification performance of the Boolean kernels proposed in this paper. In the future, we will focus our research on how to learn this parameter from training dataset automatically.

References

[1] K. Sadohara, "Learning of Boolean functions using support vector machines", *Proceedings of the International Conference on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence 2225*, Springer, 2001, pp.106-118.

[2] K.Sadohara, "On a capacity control using Boolean kernels for the learning of Boolean functions", *Proceedings of IEEE International Conference on Data Mining*, IEEE Computer Society, 2002, pp.410-417.

[3] Tom M. Mitchell, "Machine Learning", McGraw-Hill Press, 1997.

[4] P. Domingos, "The role of Occam's razor in Knowledge Discovery", *Data Mining and Knowledge Discovery*, 1999, pp.409-425.

[5] N. Cristianini, J. Shawe-Taylor, "An Introduction to Support Vector Machines", Cambridge Press, 2000.

[6] B. Scholkopf, A. J. Smola, "Learning with Kernels", MIT Press, 2002.

[7] M. Kearns, M. Li, L. Pitt, L. Valiant, "On the Learnability of Boolean Formulate", in *Proceedings of ACM Symposium on Theory of Computing*, 1987, pp.285-25.

[8] Rocco A. Servedio, "On Learning Monotone DNF under Product Distributions", *Lecture Notes in Computer Science*, Vol. 2111, Springer 2001,

[9] A. Kowalczyk, Alex J. Smola, Robert C. Williamson, "Kernel Machines and Boolean Functions", *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001, 439-446.

[10] R. Khardon, D. Roth, R. Servedio, "Efficiency versus Convergence of Boolean Kernels for Online Learning Algorithms", *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001, pp. 423-430.

[11] J. Brank, M. Grobelnik, N. Milic-Frayling, D. Mladenic, "Feature selection using support vector machines", *Proceedings of the Third International Conference on Data Mining Methods and Databases for Engineering*, Finance, and Other Fields, 2002, pp.25-27.

[12] Fabrizio Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys*, 2002, 34(1), pp.1-47.