Feature Selection by Approximating the Markov Blanket in a Kernel-Induced Space

Qiang Lou and **Zoran Obradovic** [1]

Abstract. The proposed feature selection method aims to find a minimum subset of the most informative variables for classification/regression by efficiently approximating the Markov Blanket which is a set of variables that can shield a certain variable from the target. Instead of relying on the conditional independence test or network structure learning, the new method uses Hilbert-Schmidt Independence criterion as a measure of dependence among variables in a kernel-induced space. This allows effective approximation of the Markov Blanket that consists of multiple dependent features rather than being limited to a single feature. In addition, the new method can remove both irrelevant and redundant features at the same time. This method for discovering the Markov Blanket is applicable to both discrete and continuous variables, whereas previous methods cannot be used directly for continuous features and therefore are not applicable to regression problems. Experimental evaluations on synthetic and benchmark classification and regression datasets provide evidence that the new feature selection method can remove useless variables in low and in high dimensional problems more accurately than existing Markov Blanket based alternatives.

1 INTRODUCTION

Selecting appropriate features is an important step in the data mining process whose objectives include providing more accurate and more efficient prediction as well as better understanding of data distribution. Feature selection approaches can be broadly categorized into a wrapper model [6, 7, 10] and a filter model [12, 14]. The wrapper model combines the learning method and feature selection method, which is computationally expensive and is often impractical for datasets with a large number of features. The filter model separates the feature selection from the learning process such that the results of the feature selection step are independent of the learning algorithm and are used for model learning in a follow up step.

This paper focuses on the filter model where information theory is used to find a minimum subset of the most informative features by searching the so called Markov Blanket. The Markov Blanket of a variable is regarded as a set of variables which can shield this variable from other variables. The key idea of feature selection using the Markov Blanket is to eliminate a feature for which we can find the Markov Blanket in the remaining features. Such a feature selection process is demonstrated to result in a theoretically optimal set of features [4]. Feature selection methods using the Markov Blanket can be categorized into test-based, network structure learning-based and approximate methods.

Existing test-based Markov Blanket feature selection methods [8, 16] are all using a 'Growing - Shrinking' (GS) [5] approach for discovering the Markov Blanket. In the growing phase of this approach, all features belonging to the Markov Blanket and possibly some false features enter the Markov Blanket. Then, in the shrinking phase, all features in the current Markov Blanket are checked again to remove the false features introduced at the growing phase. In both phases, conditional independence testing is used to judge if a feature belongs to the Markov Blanket or not. However, such conditional independence test-based method requires that the sample has a large number of instances to ensure the reliability of the independence test. Another limitation of test-based feature selection algorithms is that they are usually too aggressive in removing features [11].

In a structure learning-based method, a heuristic Bayesian network structure learning is performed and then the Markov Blanket is discovered corresponding to the learned structure [11]. In such an approach, to restrict search space, two heuristics (called 'sparse candidate' and 'screen-based') are proposed for selecting the promising candidates. However, the Bayesian network structure is learned using heuristic methods as the optimization here is a very hard problem. These heuristics combine locally optimal structures, which results in the learned structure that is not a global optimal solution. An additional limitation of such approaches is that learning network structure could be computationally prohibitively expensive in the presence of a larger number of features.

In an approximate Markov Blanket method called FCBF, the redundant features were eliminated in a potentially relevant subset obtained by excluding the irrelevant features based on the correlation to the target variable [15]. In this approach, symmetrical uncertainty is used to measure the relation between variables. For a pair of features, FCBF measures their symmetrical uncertainty and also the symmetrical uncertainty between either of them and target variable. If the measured value between these two variables is bigger than the measured value between one of them and the target variable, the variable with larger symmetrical uncertainty to the target is regarded as the Markov Blanket of the other variable which is removed. FCBF assumes the Markov Blanket of a feature has only one feature, since it is based on pairwise comparison. Such an approach is often too restrictive in practical situations as illustrated in the results section of this article. In addition we found that FCBF is too aggressive in eliminating features, since it gives too much priority to dominant features.

The feature selection method proposed and evaluated in this study approximates the Markov Blanket without relying on the conditional independence test or network structure learning. This is achieved by efficiently measuring dependence among variables according to Hilbert-Schmidt Independence Criterion and using this to find effec-

[1] Center for Information Science and Technology, Temple University, Philadelphia, PA 19122, USA. email: qianglou@temple.edu, zoran@ist.temple.edu

tively an approximation of the Markov Blanket that consists of multiple dependent features rather than being limited to a single feature as in FCBF [15]. In addition, the new method can remove both irrelevant and redundant features at the same time. An additional strength of this method for discovering the Markov Blanket is that it is applicable to both discrete and continuous variables, whereas previous methods cannot be used directly for continuous features and targets and therefore are not applicable to regression problems.

2 MEASURING DEPENDENCE AMONG VARIABLES

Feature selection requires using an appropriate correlation or dependence measure to evaluate the relationship between features and the target variable, or between features and features. In our study we measure the dependence among variables in an appropriate kernel space. In this section, we will first describe a dependence measure called Hilbert-Schmidt Independence Criterion (HSIC) and then discuss the reason why we use it as the basis measure for the Markov Blanket discovery.

A Hilbert space F of functions in which pointwise evaluation is a continuous linear functional is called a Reproducing Kernel Hilbert Space (RKHS) [12]. In other words, in RKHS for an arbitrary feature set X, there is a mapping $\varphi : X \rightarrow F$ to a Hilbert space F such that $< \varphi(x), \varphi(x') >= k(x, x')$, where k is a unique positive definite kernel [3]. Let X and Y be sets drawn from some joint probability distribution Pr_{xy}. Let F be the RKHS on X with, $k : X \times X \rightarrow \Re$ and $\varphi : X \rightarrow F$ be the corresponding kernel and feature map. Similarly, let G be the RKHS on Y with kernel ℓ and feature map ψ. Then, the cross-covariance [9] C_{xy}: $G \rightarrow F$ is defined as

$$C_{xy} = E_{xy}[(\varphi(x) - \mu_x) \otimes (\psi(y) - \mu_y)]$$

$$= E_{x,y}[\varphi(x) \otimes \psi(y)] - \mu_x \otimes \mu_y$$

where $\otimes$ is the tensor product. Then, the Hilbert-Schmidt Independence Criterion (HSIC) is defined as the square of the Hilbert-Schmidt norm of the cross-covariance operator. A kernel version of HSIC [3] is computed as:

$$HSIC(F, G, \Pr_{xy}) = ||C_{xy}||^2_{HS}$$

$$= Exx'yy'[k(x, x')\ell(y, y')] + Exx'[k(x, x')]Eyy'[\ell(y, y')]$$

$$-2Exy[Ex'[k(x, x')]Ey'[l(y, y')]]$$

Here, we regard $E_{x,x',y,y'}$ as the expectation of two pairs (x, y) and (x', y') which are independent to each other and both drawn from Pr_{xy}.

Given a sample Z drawn from the distribution Pr_{xy}, an empirical estimate of HSIC [3] is :

$$HSIC(F_i, G, Z) = (m - 1)^{-2} trHKHL_c$$

where K and L_c are the kernel matrices of the feature F_i and the target variable C respectively, m is the number of instances and $H_{ij} = \sigma_{ij} - m^{-1}$ is used to center the features and targets in the feature space.

HSIC can detect any dependence between two variable sets X and Y by using a universal kernel (such as a Gaussian Kernel). It has been proved in [3] that $||C_{xy}||^2_{HS} = 0$ if and only if x and y are independent to each other. This is the direct motivation why we choose HSIC to measure the dependence. Actually, in this paper, we use $HSIC(F, G, Z)$ to replace the $HSIC(F, G, \Pr_{xy})$ to measure the independence between two variable sets. The reason is that $HSIC(F, G, Z)$ is easy to compute and actually it is concentrated as previously proven [3] by showing that with probability at least $1 - \sigma$

$$|HSIC(F, G, Pr_{xy}) - HSIC(F, G, Z)| \leq \sqrt{\frac{\log(6/\sigma)}{\alpha^2 m}} + \frac{C}{M}$$

where $\alpha^2 > 0.24$, $\sigma > 0$ and C is the constant.

There are three more reasons why in the proposed algorithm HSIC is used as the measure of dependence among variables. First, HSIC measures dependence in the high dimensional kernel space and so can detect any dependence between two variable sets with a universal kernel, such as RBF kernel, which is not possible with previously used measures. Second, HSIC can measure the dependence between both discrete and continuous variables. In contrast, most measures previously used to find the Markov Blanket are entropy-based, and so they are not directly applicable in datasets with a continuous variable. Third, HSIC is easy to compute from the kernel matrices without density estimation.

3 APPROXIMATING THE MARKOV BLANKET

Let F be the whole set of features. The Markov Blanket MB_i of feature F_i ($MB_i \subset F$, ($F_i \notin MB_i$)) is the set of features with a property that F_i is conditionally independent of the remaining features U and the target C. More formally, set MB_i is called the Markov Blanket of F_i iff:

$$P(U, C|F_i, MB_i) = P(U, C|MB_i)$$
$$\text{where} : U = F - \{F_i\} - MB_i$$

If MB_i is the Markov Blanket of F_i, then the prediction model learned without considering F_i is as accurate as the model learned using all features F.

It is often difficult to find the exact Markov Blanket for a given feature. To address this problem we propose a novel method of finding an approximate Markov Blanket for F_i. We then use this method to develop a feature selection algorithm based on the discovered approximate Markov Blanket.

Given a set of features we can easily check if it is the Markov Blanket MB_i of feature F_i. However, evaluating all subsets of F for this property is prohibitively costly. To reduce the search cost we will evaluate some candidate subsets, as proposed in the following subsection.

3.1 Identification of the Markov Blanket candidates

Instead of searching for the exact Markov Blanket for a feature, in practice it is appropriate to determine an approximate Markov Blanket that can be used for removing this feature with little useful information lost. Intuitively, if MB_i is the Markov Blanket for feature F_i, the features in MB_i are more dependent to F_i than those features which are not in MB_i [1]. Therefore, we can choose a subset of k features which are strongly dependent to F_i as the candidate Markov Blanket of F_i. Then, for each feature, we only need to evaluate its candidate Markov Blanket rather than all possible subsets in the remaining features to see if such a candidate Markov Blanket is sufficiently accurate to be regarded as the Markov Blanket.

The problem with this reasoning is how to find efficiently the Markov Blanket candidate for each feature. The naive method [4] to find the set of k features which are most dependent to F_i requires computing the dependence $HSIC(F_i, F_j) = (m-1)^{-2} trHK_iHK_j$ for all pairs of features F_i and F_j (here, K_i and K_j are the kernel matrices of feature F_i and F_j respectively). This is clearly computationally too expensive for applications in high dimensional datasets. Instead, for F_i we will compute an approximate Markov Blanket candidate MB_i whose each feature F_{MB_i} satisfies:

$$F_{MB_i} = \arg\max_{F_B} HSIC(K_{F_B}, K_i),$$

$$\text{where} : F_B \in B_i - MB_i \cup \{F_{MB_i}\}$$

Here, B_i is a set of features which are more dependent to the target variable C than F_i, and K_{F_B} and K_i are kernel matrices of feature F_B and F_i respectively.

Observe that we tend to find the approximate Markov Blanket for F_i in the features which are more dependent with the target variable C than F_i is.

To find the Markov Blanket Candidate for F_i, we measure dependence of each feature in F to target C and, for each feature F_i, consider only a subset B_i of features that are more dependent to C than F_i is. Here, dependence of feature F_i to C is measured as $HSIC(F_i, C) = (m-1)^{-2} trHK_iHL_c$, where$K_i$ and L_c are the kernel matrices of feature F_i and target variable C. We approximate the Markov Blanket Candidate of F_i as the set of *k* features from set B_i that are most dependent to F_i. For the features whose corresponding set B_i has less than *k* features, we choose all features in B_i as the Markov Blanket candidate of F_i.

We emphasize that quality of the Markov Blanket Candidate obtained by the proposed method depends on choice of *k* which should not be too large or too small (too large *k* could include features that are irrelevant while too small *k* could result in an incomplete Markov Blanket). However, our experiments reported in Section 5 provide evidence that this is not a serious limitation in practice since the proposed method was quite robust over a large range of choices of *k*.

3.2 Screening the Markov Blanket candidates

Let MB_i be the Markov Blanket candidate of feature F_i, found as explain in section 3.1. We say that MB_i passes the dependence-based screening test and is regarded as an actual approximation of the Markov Blanket if it satisfies the following two conditions:

$$1.HSIC(MB_i, C) > HSIC(MB_i \cup F_i, C)$$
$$2.HSIC(MB_i, C) > HSIC(F_i, C), \text{and}$$
$$HSIC(MB_i, F_i) > HSIC(F_i, C)$$

where C is the target variable and HSIC(X, Y) is defined as the dependence measure between two variable sets X and Y.

We remove the feature whose Markov Blanket candidate passes this screening test. In contrast to a previous work [3], we remove both irrelevant and redundant features at the same time rather than separating into two steps. This is appropriate since an independent irrelevant feature F_i always satisfies the first test condition while a dependent irrelevant F_i will satisfy the second condition as in such a case F_i is irrelevant to C resulting in $HSIC(F_i, C)$ smaller than $HSIC(MB_i, F_i)$. Similar, condition 2 ensures that the redundant feature is removed, since the corresponding MB_i of such F_i can subsume the information this feature have about the target variable. $HSIC(MB_i, F_i) > HSIC(F_i, C)$ implies F_i is more dependent to MB_i than to C; $HSIC(MB_i, C) > HSIC(F_i, C)$ means MB_i is more dependant to C than F_i and ensures MB_i has more deterministic information to the target variable C than F_i does.

4 FEATURE SELECTION ALGORITHM

The optimal feature selection using the Markov Blanket is based on removing a feature for which we can find the Markov Blanket in the remaining features. Instead, in our computationally efficient method, we remove the feature for which we find an approximate Markov Blanket. According to the approximate Markov Blanket construction described in Section 3, we propose the following independence-based feature selection algorithm that will be called Hilbert-Schmidt Markov Blanket method (HSMB).

For each F_i in the whole feature set F, this algorithm computes $HSIC(F_i, C)$ which is the dependence between F_i and target variable C and then sorts features into a list S in descending order based on the measured dependence. Then, for each feature F_i, the algorithm constructs set B_i consisting of the features which are located before F_i in the list S. Then, HSMB finds the Markov Blanket candidate MB_{can} of F_i which is exactly the *k* features in the set B_i that are most dependent to F_i. If MB_{can} passes the screen test, it is regraded as the Markov Blanket of feature F_i. Therefore, the algorithm will remove such feature F_i from the sorted list S. In this way in a single pass through the list of features, HSMB removes all features for which the algorithm finds the approximate Markov Blanket in the remaining set of features. No multi-iteration is needed in HSMB algorithm. The HSMB algorithm is summarized in Algorithm 1.

Algorithm 1 HSMB

```
Input: $F = F_1, F_2, ..., F_N, C$ // training data set with N features
and target C
Output: MB // a set of selected features

for i=1 to $N$ do
  calculate $HSIC(F_i, C)$;
  insert $F_i$ into list S based on $HSIC(F_i, C)$;
end for
for i = 1 to $N$ do
  $MB_{can} = \emptyset$
  $B_i = \{F_j | F_j$ is before $F_i$ in S$\}$
  for j = 1 to $k$ do
    //usually, $1 \leq$ k $\leq 5$
    $F_c = \arg\max_{F_{B_i} \in B_i} HSIC(K_{F_{B_i}}, K_i)$;
    $MB_{can} = MB_{can} + F_c$;
    remove $F_c$ from $B_i$
  end for
  if $F_i$ and $MB_{can}$ pass the screen test then
    remove $F_i$ from S;
    $MB = S$;
  end if
end for
```

The main cost of the HSMB algorithm is in computing HSIC values which has complexity of O(M^2) in terms of the number of instances M. This is better than the cost of other kernel-based methods which usually have O(M^3) complexity. Cost of HSMB in terms of the number of features N is smaller. However, for each feature we have to find the candidate Markov Blanket in which there are *k* features. This requires searching for *k* most dependent features in the set

Table 1. Selected features on synthetic classification data ($A0_1$ and $B0_1$ mean the first redundant feature of A0 and B0 respectively).

Data Sets	Optimal Sets	GS	FCBF	BAHSIC	HSMB
Corral-7	A0,A1,B0,B1	R,B1,A0	R,A0	R,A0,A1,B0,B1	R,A0,A1,B0,B1
Corral-46	A0,A1,B0,B1	B0,A0	A0,A1,B0,B1	A0,$A0_1$,B1,A1	A0,A1,B0,B1
Corral-rel-7	A0,A1,B0,B1	R,A0,B1	R,A0,B0	R,A0,B0,A1,B1	R,A0,A1,B0,B1
Corral-rel-46	A0,A1,B0,B1	A0,B1	A0,B0	A0,$A0_1$,B0,$B0_1$	A0,A1,B0,B1

Table 2. Selected features on synthetic regression data. Only BAHSIC and HSMB can work in regression problem. For BAHSIC, we choose the same number of features as selected in HSMB automatically.

Data sets	Optimal Sets	BAHSIC		HSMB	
		Selected Features	R-square	Selected Features	R-square
Regression-22	x1,x2	x1,x2	0.86	x1,x2	0.86
Regression-38	x1,x2	x1,x1	0.69	x1, x2	0.86
Regression-138	x1, x2	12 features	0.54	12 features	0.75

B_i (features before F_i in the list S) to find the Markov Blanket Candidate. Hence, to select the optimal subsets MB, the algorithm takes $\mathrm{O}(p * N)$ steps, where p is the number of features before a certain feature F_i in the list S. In the worst case, p becomes N, and then the cost of the algorithm is $\mathrm{O}(N^2)$. However, p is a small constant when enough features are removed resulting in $\mathrm{O}(N)$ best case complexity. This best case scenario corresponds to many high dimensional datasets where we are likely to remove most features.

5 EXPERIMENTAL RESULTS.

The accuracy of the proposed feature selection algorithm is evaluated on a variety of synthetic and benchmark datasets as reported in this section. Synthetic data is used to compare the new algorithm to the optimal solution [15]. The new algorithm is also compared to the GS algorithm [5] which is a test-based Markov Blanket discovering method, FCBF [15] which uses an approximate Markov Blanket method and BAHSIC [13] based on HSIC feature selection without Markov Blanket approximation. In experiments reported in sections 5.1-2 the *k* parameter of our algorithm was fixed to 3 and the influence of different *k* was explored in section 5.3.

As a learning method for all feature selection algorithms, we used SVM with a Gaussian kernel and with δ set to be the median distance between points in the sample [2]. In FCBF, GS and HSMB, the number of features was determined automatically. BAHSIC is not providing such an option as it ranks all features. So, for a meaningful comparison, in BAHSIC we report the results using the largest of the number of features automatically selected by FCBF, GS and HSMB.

5.1 Results on synthetic data

5.1.1 Classification on synthetic datasets

The first two datasets called Corral-7 and Corral-46 were used previously to prove the strength of FCBF [15]. Corral-7 consists of six Boolean features, A0, A1, B0, B1, R, I and a Boolean class Y defined by $Y = (A0 \wedge A1) \vee (B0 \wedge B1)$. Among these six features, feature A0, A1, B0 and B1 are independent to each other, feature I is uniformly random, and feature R matches the class Y 75 percents of the time. The optimal subset for the Corral-7 data set includes A0, A1, B0 and B1, whereas I is the irrelevant feature and R is the redundant feature. Corral-46 includes the same A0, A1, B0 and B1 as Corral-7, but it also has 14 additional irrelevant and 28 additional redundant features. We also evaluated methods on Corral-rel-7 and Corral-rel-46 datasets that are similar to Corral-7 and Corral-46, respectively, except that A0, A1, B0 and B1 in Corral-rel-7 and Corral-rel-46 are not independent to each other any more. In the last two datasets, A0 matches A1 60 percents of the time, and the same level of dependence is introduced between B0 and B1. The feature selection results on these datasets are shown in Table 1. In all four experiments HSMB selected all four relevant features. In two of these experiments (Corral-46 and Corral-rel-46), it found the optimal set and in remaining cases (Corral-7 and Corral-rel-7) it only included one extra redundant attribute. This was better than any of the alternative algorithms. In particular, for the Corral-7 and Corral-rel-7 datasets, both GS and FCBF removed too many features, which is consistent with the previous discussion that these two methods are too aggressive in removing features. In Corral-rel-7 and Corral-rel-46, FCBF regarded A1 and B1 as redundant features of A0 and B0 respectively, even though they should be in the optimal set. BAHSIC did not work well on Corral-46 and Corral-rel-46, since it applies traditional greedy backward feature selection which is not efficient in removing redundant features.

5.1.2 Regression on synthetic datasets

Three synthetic datasets were used for evaluation on regression problems. The first dataset that we call Regression-22 was previously used to demonstrate the advantages of BAHSIC [13]. Here, the label y is generated as $y = x1 * e^{-x1^2 - x2^2} + \varepsilon$, where ε is random Gaussian noise. Regression-22 consists of x1, x2 and 20 additional irrelevant features. Based on Regression-22, we generate two more datasets, Regression-38 and Regression-138. Regression-38 includes all features used in Regression-22 and 16 additional redundant features. Among the 16 redundant features, for either x1 or x2, there are 8 redundant features matching x1 and x2 at 9/16, 10/16, ... , 16/16 of times, respectively (we use random values for those which do not match). Regression-138 includes all features used in Regression-38 and 100 additional irrelevant features. Since BS and FCBF methods can not be applied directly to regression problems, in these experiments we compared the results of our method to BAHSIC. The results of these experiments provide evidence that the new method is equally accurate on Regression-22 and is much more appropriate in the presence of redundant features and when the number of irrelevant features is larger (R-square accuracy and selected features shown in Table 2). For BAHSIC, we choose the same number of features as selected in HSMB automatically to build the prediction model and measure the R-square accuracy.

5.2 Results on benchmark data.

5.2.1 Classification results on benchmark problems.

The number of features, instances and classes of 12 benchmark datasets for the classification experiments are summarized in Table 3. Here, the 3-class Lung-cancer dataset is converted to three 2-class problems and the average accuracy of these three 2-class problems is reported. In the Promoters dataset each categorical feature is converted into 4 binary features resulting in 228 (57*4) features. In the Isolet dataset, the original 10-class problem is converted to a binary classification problem by regarding labels which are not bigger than 13 as class 0 and the remaining as class 1.

Table 3. Properties of benchmark datasets for classification.

Datasets	Features	Instances	Classes
BC-wisconsin	9	683	2
Hepatitis	18	112	2
German	24	1000	2
Wdbc	30	569	2
wpbc	33	194	2
Lung-Cancer	56	32	3
COIL2000	85	5822	2
High	Dimensional	Data	
Musk2	166	6598	2
Promoters	228	106	2
Madelon	500	2000	2
Isolet	617	1559	26
Arcene	10000	100	2

Table 4. Classification accuracy '%' on benchmark datasets ('All' means using all features).

Datasets	All	GS	FCBF	BAHSIC	HSMB
BC-Wisconsin	**96.8**	**96.8**	95.6	**96.8**	**96.8**
Hepatitis	82.6	83.0	86.9	78.3	**89.8**
German	**100**	**100**	**100**	**100**	**100**
Wdbc	96.7	97.0	96.2	95.0	**97.3**
wpbc	80.0	72.0	75.0	78.0	**81.5**
Lung-Cancer	67.9	62.0	65.0	**70.4**	**70.4**
COIL2000	93.7	94.0	**94.4**	90.2	**94.4**
High Dimensional	Data				
Musk2	86.0	88.0	89.2	89.1	**92.4**
Promoters	86.4	88.5	97.6	94.7	**98.6**
Madelon	56.0	63.3	61.5	62.0	**70.1**
Isolet	75.1	78.4	78.5	78.0	**81.3**
Arcene	59.4	60.2	62.4	64.2	**70.6**

The results of 12 benchmark classification problems are summarized in Table 4 and Table 5. Each result listed in these two tables is an average of 5 repeated experiments. Table 4 shows the predictive accuracy. Here, the leave one out method is applied to the datasets with less than 300 instances, in order to get stable results of SVM predictors. Other results reported in these tables are obtained by average results of 5 repeated experiments. The obtained results (Table 4) provide evidence that HSMB outperforms alternative methods in accuracy over a variety of benchmark datasets. On Lung-Cancer, Arcene, Promoters and wpbc evaluations, the GS algorithm was less accurate as these datasets have a fairly small number of instances which makes a conditional dependence test unreliable.

Table 5. Number of selected feature on benchmark datasets for classification (HSIC method is not shown since it cannot automatically select the optimal set).

Datasets	GS	FCBF	HSMB
BC-wisconsin	8	8	8
Hepatitis	4	3	10
German	20	21	22
Wdbc	6	3	15
wpbc	3	2	6
Lung-Cancer	3	5	3
COIL2000	5	6	8
High	Dimensional	Data	
Musk2	4	2	6
Promoters	15	13	17
Madelon	4	2	38
Isolet	4	4	10
Arcene	15	13	113

The number of features selected by each feature selection method is reported in Table 5. All methods reduced the number of features significantly. However, the GS and FCBF algorithms were too aggressive in reducing features. In GS, this problem is due to unreliability of the conditional dependence test for a small sample. The reason FCBF tends to remove too many features is that it gives too much priority to features that are highly correlated with the target.

Our results show that the proposed algorithm works well on both low and high dimensional data sets. In the high dimensional data set with a large number of irrelevant features, HSMB was effective in selecting the dominant features from which accurate predictors were built. In contrast, BAHSIC had high error in high dimensional evaluations.

5.2.2 Regression results on benchmark problems.

Six benchmark datasets were used for the evaluation of the new algorithm for regression problems. Available benchmark datasets for regression were low dimensional. Therefore, for high dimensional evaluation we have created three datasets based on the Housing benchmark dataset. Housing-100 includes the original 13 features of Housing dataset with 48 additional redundant features and 52 irrelevant features. Among the 48 redundant features, for each feature in the optimal set selected by HSMB from Housing, there are 8 relevant features which match the selected feature 9/16, 10/16, ... , 16/16 of time. Housing-500 includes the features used in Housing-100 and additional 400 irrelevant features. Similar, Housing-1000 includes all features of Housing-100 and 900 additional irrelevant features.

For each regression benchmark dataset, we performed the experiments following the same procedure as that used in classification. Our method is compared only to BAHSIC as other two methods are specific for classification. Again, in BAHSIC the same number of features is used as returned by our method. The R-square accuracies of an SVM on the subsets selected by each of two feature selection methods on benchmark datasets are reported in Table 6. The obtained results were consistent results with classification results of section 5.2.1. In general, HSMB outperformed BAHSIC in these regression problems. In low dimensional experiments, the predictor using full features was the most accurate since there were almost no irrelevant features in these benchmark datasets. However, in high dimensional experiments with many irrelevant features (Housing-100, Housing-500 and Housing-1000), the dominant features selected by the pro-

Table 6. Regression accuracy (R-square) on benchmark datasets. (BAHSIC is using the same number of features as selected by HSMB.)

Data Set		All Features		BAHSIC	HSMB	
Name	# instances	R-square	# Features	R-square	R-square	Selected Features
Cpu-performance	209	**0.575**	6	**0.575**	**0.575**	5
Auto-mpg	392	**0.745**	7	0.70	0.742	5
Concrete	1030	**0.880**	8	0.79	0.86	5
Housing	506	**0.634**	13	0.534	0.613	6
Aerosol	2000	**0.756**	14	0.61	0.714	6
Auto-mobile	159	**0.492**	18	0.391	0.398	5
High Dimensional	Data					
Wpbc	194	0.989	33	0.982	**0.997**	11
Housing-100	506	0.601	113	0.520	**0.610**	9
Housing-500	506	0.453	513	0.531	**0.593**	23
Housing-1000	506	0.364	1013	0.482	**0.574**	82

posed method were a much better choice.

5.3 Influence of parameter *k*

The quality of an HSMB solution depends on the number *k* of features used for an approximation of the Markov Blanket (as described in Section 3.1). Sensitivity of HSMB to *k* is evaluated on benchmark datasets. Our experimental results suggest that the results of HSBM method are quite stable over a large range of *k* values and that for most datasets 2 or 3 is a good choice for the value of *k* . For $k = 1$ to 5, classification accuracy on some benchmark datasets is shown in Table 7. These experimental results are consistent with what we describe in Section 3.1 that *k* should not be too large or too small. The results on remaining datasets are similar and are omitted here for the lack of space.

Table 7. Classification accuracy (%) of HSMB for different values of *k*.

Datasets	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
BC-wisconsin	96.2	93.8	96.8	**97.1**	96.7
Hepatitis	87.0	**89.8**	**89.8**	86.3	85.8
German-numeric	**100**	**100**	**100**	**100**	**100**
Wdbc	93.9	**97.6**	97.3	94.4	95.4
wpbc	79.4	**81.5**	**81.5**	79.5	76.9
Lung-Cancer	67.9	67.9	**70.4**	**70.4**	65
COIL2000	93.7	**94.4**	**94.4**	92.6	92.6
Musk2	90	89.2	**92.4**	86	86
Promoters	98.1	98.1	**98.6**	96.2	95.3

6 CONCLUSIONS

In this paper we proposed a new Markov Blanket based filter method for feature selection. This method, called HSMB, uses Hilbert-Schmidt Independence Criterion (HSIC) as the measure of dependence in finding the Markov Blanket. We restrict the search space by approximating the Markov Blanket Candidate of feature F_i as the *k* most dependent features in the set B_i consisting of the features before F_i in the sorted list S in which all features are ordered by their dependence to the target variable. Results on synthetic and benchmark datasets provide evidence that HSMB can select better subsets of features than the alternative Markov Blanket based methods. The new method is applicable to both low dimensional and high dimensional classification and regression problems. To improve the proposed method further, we will conduct additional theoretically and empirically research related to selection of features for the Markov Blanket Candidates approximation and for optimizing the value of *k* automatically.

REFERENCES

[1] Kozlov, A. V., Singh, and J. P. Sensitivities, An Alternative to Conditional Probabilities for Bayesian Belief Networks In Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence., 1995, pp. 376-385.

[2] Scholkopf, B. and Smola, A. Learning with Kernels. Cambridge, MA, MIT Press, 2002.

[3] Gretton, A., Bousquet, O., Smola, A., and Scholkopf, B. Measuring Statistical Dependence with Hilbert-Schmidt Norms In Algorithmic Learning Theory 2005, 63-78.

[4] Koller, D. and Sahami, M. Toward Optimal Feature Selection, In International Conference on Machine Learning, 1996, 284-292.

[5] Margaritis, D., and Thrun, S. Bayesian Network Induction via Local Neighborhoods, In Neural Information Processing Systems (NIPS) 1999, 12:505-511.

[6] Kohave, R. and John, G. Warppers for Feature Subset Selection, Artificial Intelligence, 1997, 1-2: 273-324.

[7] Guyon, I. and Elisseeff A. An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 2000, 3:1157-1182.

[8] Tsamardinos, I. and Aliferis, C. Towards Principled Feature Selection: Relevancy, Filters, and Wrappers, In Ninth International Workshop on Artificial Intelligence and Statistics, 2003.

[9] Fukumizu, K., Bach, F. R., and Jordan, M. I. Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces, Journal of Machine Learning Research 2004, 5, 73-99.

[10] Kim, Y., W. Nick Street, and Menczer, F. Feature Selection for Unsupervised Learning via Evolutionary Search, In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, 365-369.

[11] Shen J., Li, L. and Wong, W . Markov Blanket Feature Selection for Support Vector Machines, In AAAI Conference on Artificial Intelligence 2008.

[12] Song, L., Smola, A., Gretton, A. and Borgwardt, K. M. A Dependence Maximization View of Clustering, In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007.

[13] Song, L., Smola, A., Gretton, A., Borgwardt, K. M., and Bedo, J. Supervised Feature Selection via Dependence Estimation, International Conference on Machine Learning 2007.

[14] Yu, L., and Liu, H. Feature Selection for High-dimensional Data: A Fast Correlation-based Filter Solution, In Proceedings of the twentieth International Conference on Machine Learning, 2003, 856-863.

[15] Yu, L. and Liu, H. Efficient Feature Selection via Analysis of Relevance and Redundancy, Journal of Machine Learning Research 2004, 5, 1205-1224.

[16] Yaramakala, S. and Margaritis, D. Speculative Markov Blanket Discovery for Optimal Feature Selection. International Conference on Machine Learning 2005.