Discovering Highly Informative Feature Set Over High Dimensions

Chongsheng Zhang, Florent Masseglia, Xiangliang Zhang

Discovering Highly Informative Feature Set Over High Dimensions

Chongsheng Zhang*
Henan University
475004 Kaifeng, China
Email: Chongsheng.Zhang@yahoo.com

Florent Masseglia*
Zenith Team, INRIA
34095 Montpellier, France
Email: Florent.Masseglia@inria.fr

Xiangliang Zhang
MINE Team, KAUST
Thuwal 23955-6900, Saudi Aribia
Email: Xiangliang.Zhang@kaust.edu.sa

***Abstract*—For many textual collections, the number of features is often overly large. These features can be very redundant, it is therefore desirable to have a small, succinct, yet highly informative collection of features that describes the key characteristics of a dataset. Information theory is one such tool for us to obtain this feature collection. With this paper, we mainly contribute to the improvement of efficiency for the process of selecting the most informative feature set over high-dimensional unlabeled data. We propose a heuristic theory for informative feature set selection from high dimensional data. Moreover, we design data structures that enable us to compute the entropies of the candidate feature sets efficiently. We also develop a simple pruning strategy that eliminates the hopeless candidates at each forward selection step. We test our method through experiments on real-world data sets, showing that our proposal is very efficient.**

I. Introduction

Feature selection is the task of selecting interesting or important features, and removing irrelevant or redundant ones. It has been widely used in many application fields, such as discriminative gene selection [3] and text categorization [8]. Before selecting the features, an interestingness measure should be defined to assess the significance of the features or featuresets (i.e. feature sets). For instance, if the interestingness is measured by a given utility function, we select feature sets with the best (highest or lowest) utility scores. Given the measurement for the feature sets, one can design efficient algorithms to search for the best feature sets, where wrapper and filter [5] are the commonly used methods.

Besides the interestingness measures, feature selection is closely related to the characteristics of the data itself. Feature selection techniques for labeled data are rather different from those for unlabeled data [1]. For unlabeled data, we do not have class as references and feature selection always depends on the applications and tasks. For example, when choosing the top-k most important keywords from large documents, the tf-idf method [6] can be used to measure the weight of each keyword; when finding a feature set that contains the most information, one can resort to information theory based feature selection methods [2], [4]. Example 1 addresses the problem of feature selection for document retrieval.

Example 1: We would like to retrieve documents from table I, in which the columns of $\{O_1, ..., O_{10}\}$ are 10 documents, and the attributes of $\{A, B, C, D, E\}$ are some features (key words) in the documents, where the value "1" means that the feature is contained in the document, and "0" not. It is easy to find that (D, E) is a frequent featureset, because features D and E occur together in nearly every document. However, it provides little help for document retrieval. By contrast, (A, B, C) is an infrequent featureset, as its member features rarely or never appear together in the data. And it is troublesome to summarize the value patterns of featureset (A, B, C). Providing it with the values $< 1, 0, 0 >$ we could find the corresponding document O_3; similarly, given the values $< 0, 1, 1 >$ we will have the according document O_6. Although (A, B, C) is infrequent, it contains lots of useful information which is hard to summarize. We call it an informative feature set.

TABLE I
Features in the documents

Feat.	Documents									
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
A	1	1	1	1	1	0	0	0	0	0
B	0	1	0	0	1	1	0	1	0	1
C	1	0	0	1	0	1	1	0	1	0
D	1	0	1	1	1	1	1	1	1	1
E	1	1	1	1	1	1	1	1	1	1

From the above example, we can see that it is useful to discover informative feature sets. Indeed, information theory provides strong support to measure the informativeness of the feature sets. But for unlabeled data, it requires computing the joint probability not just for the case when all the features have appeared together, but also for all the other cases when some features have appeared in the transactions but others have not. This is an exhaustive work. To tackle this problem, the authors in [4] proposed an effective and efficient method for selecting the informative feature sets over low-dimensions. However, this algorithm does not scale well for high-dimensions.

In this paper, we investigate the problem of efficient discovery of most informative feature set over unlabeled high-dimensional data. This is a very challenging task. First, there are $\frac{d!}{k! \times (d-k)!}$ candidate feature sets having k features, k is a user-specified parameter denoting the size of the feature set

*This work has been conducted and funded when the first two authors were members of the AxIS team at INRIA Sophia Antipolis.

that s/he expects, and d is the data dimension. Because of the high dimensions, the number of candidates increases greatly w.r.t. d. Second, due to the characteristics of informative feature sets, we have to compute for each candidate feature set, the probability for every existing case of feature composition in all the transactions, and there are up to 2^k possible cases of feature combinations. If the number of transactions is n, we need as many as $n * 2^k$ computations for each candidate feature set. Overall, we need $n * 2^k * \frac{d!}{k! \times (d-k)!}$ calculations for all the candidates. This is very computation-demanding.

To solve the above problem, we propose a heuristic theory. Exploiting the fact that usually only a few features are correlated in most textual collections, our heuristic theory i) divides the features into independent and dependent sets and ii) reduces the candidate features for informative feature sets to a rather small subset. Upon this theory, we introduce a forward selection algorithm that discovers the most informative feature set based on the selected features. Moreover, we design a data structure that speeds up the computation of the entropies of the candidate featuresets. We also introduce a pruning strategy that helps filter out the unpromising candidate featuresets. Based on all our proposal, we build a framework ***IIS*** (Feature **I**ndependence based **I**nformative featureset **S**election). Experiments on real-world datasets show that our method can save 90% computation time comparing with the algorithm from the literature.

The remaining of this paper is organized as follows. In section II, we discuss the related work about feature selection. After giving the related definitions in section III, we introduce the theory for feature set mining and give our algorithm and two optimization techniques, as well as the concrete *IIS* framework in section IV. We evaluate our work in section V. Finally, we conclude and give perspectives in section VI.

II. Related Work

Feature selection is a very important tool for many data mining applications. In [5] we find a systematic review of the foundations, techniques and applications of existing feature selection methods. This paper focuses on selecting the most informative feature set over high-dimensional unlabeled data. We therefore discuss the related work referring the keywords of *informative feature set*, *high-dimensional data* and *unlabeled data*.

Informative feature set selection on unlabeled data has been studied in recent years. Knobbe et al. in [4] proposed a heuristic algorithm to extract informative featureset of size k with the highest entropy. Such featuresets are called *miki*. The proposed *ForwardSelection* algorithm performs multiple scans over the dataset. Within each scan, *miki*'s size gradually increases, by adding a new feature f to the *miki* at current iteration. They showed the advantage of *ForwardSelection* over the brute force algorithms that evaluate the entropies of all the possible subsets of size k. Afterwards, method for low entropy feature set discovery was also investigated. In [2], Heikinheimo et. al. proposed two algorithms designed to extract the sets of both low entropy and high entropy featuresets from static datasets. In [10], we extended *miki* to work on streaming data with transient features and proposed to reduce its candidates by removing similar sets according to mutual information criteria. But these methods become inefficient in high-dimensions, because we will have huge candidate feature sets to check in high-dimensions, and for each candidate we have to calculate the counts and compute the probability for all appeared feature combinations, including the ones where some features appeared but others did not. Our work aims to speed up the searching and computation for most informative feature set.

There are many approaches in feature selection for high-dimensional data. But most of them are towards labelled data. Yu and Liu [9] use the class concept as the reference and identify all predominant features which are most capable to predict class concept. The proposed method is based on discovering pair uncertainty correlation with the class concept and other features. They only considered the pair correlationship, but the cases that several features jointly decide the class concept have not been considered. Our work is about informative feature set selection over unlabeled and high dimensional data, and the task is to find the feature set with highest joint uncertainty, whereas the features can be dependent or independent of others. *FastANOVA* [11] was designed for joint association discovery, but strong association (or correlation) within a feature set does not imply a high information (i.e. joint entropy) contained in the feature set. In fact, even though the features in the set are weakly associated, their joint entropy can be very high.

III. Definitions

Preliminary Definitions. We refer to book [7] for the preliminary definitions of entropy, joint entropy of a feature set, mutual information and relevance coefficient. Let $H(X)$ denote the entropy of feature X, and $H(I)$ represent the joint entropy of featureset I. The mutual information between two discrete variables can be computed by Formular 1. The **Relevance Coefficient** (**RO**) between two features is measured by Formula 2. Features A and B are **independent** if $RO(A,B) = 0$; A and B are **mutually redundant** if $RO(A,B) = 1$.

$$MI(AB) = H(A) + H(B) - H(AB) \tag{1}$$

$$RO(A,B) = \frac{MI(A,B)}{MIN\{H(A), H(B)\}} \tag{2}$$

To relax the conditions of feature independence and redundancy, we introduce the following definition and equation to measure the relationship between two features, using a lower bound threshold ξ and a upper bound threshold δ.

Definition 1: The **relationship** between two features depends on their RO value.

$$A \text{ and } B \text{ are } \begin{cases} independent, & if\ RO(A,B) < \xi; \\ redundant, & if\ RO(A,B) > \delta; \\ relevant, & otherwise \end{cases} \quad (3)$$

Now let us define the highly/most informative feature set.

Definition 2: Given a size k, the **Highly/Most Informative Featureset** (*HI*) is the featureset that has the largest entropy value among all the candidates having size k, as given by formula 4.

$$H(HI) = MAX\{H(IS_k), IS_k \in SS_k\} \quad (4)$$

where $IS = \{I_1, I_2, ..., I_n\}$ is the set of all n possible features, $IS_k = \{I_{m_1}, I_{m_2}, ..., I_{m_k}\}$ is a featureset of size k ($k < n$), $SS_k = \{IS_k\}$ is the set of all possible featuresets of size k.

Problem Definition. In this paper, we aim at discovering the most informative featureset over high-dimensional unlabeled dataset. Given such high dimensions, our approach should reduce the search space and be as effective as possible.

IV. Informative Featureset Selection Over High Dimensions

In this section, we first introduce a heuristic theory that forms the foundation of our framework. Then we present the *MIFS* forward feature selection algorithm. We next introduce two techniques to speed up the searching and computation for highly informative featureset. Finally, we introduce the *IIS* framework.

A. The Heuristic Feature Reduction Theory

Before illustrating the theory, we first give some basic ideas and concepts. k is a user input parameter, denoting the size of the expected informative feature set. In our theory, we perform the following steps:

1) Sort the features by entropy;
2) Evaluate independence between features, and categorize them into dependent and independent sets;
3) Fetch the top-k features from the independent sets and put them into *BRFSet* (**B**rief **R**eference **F**eature **Set**);
4) Get all the features which rank higher than the last feature in *BRFSet*, and insert them into *MiniSet* (**Mini**mum candidate feature **Set**).
5) Considering some features in *MiniSet* could have correlated features, we maintain *MaxSet* to include both *MiniSet* and all the features which are correlated with features in it;

Finally, our theory limits the search space of informative feature sets to *MaxSet*, so we can perform informative feature set selection methods on *MaxSet*. We explain the reasons for choosing *MaxSet* in the following.

We first refer to the following theorem in [7].

Theorem 1: If and only if variables X_1,X_2, ..., X_{n-1},X_n are mutually independent, then

$$H(X_1, X_2, ..., X_{n-1}, X_n) = \sum_{i=1}^{n} H(X_i) \quad (5)$$

If the condition in theorem 1 is satisfied, it will be simple to compute the joint entropy. Otherwise, we claim the following inference.

Lemma 1: If S = $\{X_1$,X_2, ..X_{m-1},X_{m+1}... X_{n-1},$X_n\}$, and X_m, $1 \leq m \leq n$, is independent of any of the features in S, but the features in S are not necessarily mutually independent. Then

$$\begin{aligned} &H(X_1, X_2, ..X_{m-1}, X_m, X_{m+1}....X_{n-1}, X_n) \\ &\neq H(X_m) + H(X_1, X_2, ..X_{m-1}, X_{m+1}...X_{n-1}, X_n) \end{aligned} \quad (6)$$

Proof: The reason is that X_m is not independent of S. Let us take a small example of three variables A, B and C. In table II, the left part is the data, and the right part is the joint probability of these three features.

Judging from the joint probability of (A,B) and (A,C), we can see that A and B, A and C are independent respectively. However, this is not the case for (A,B,C). Actually, $P(A = 0, B = 1, C = 0) = 1/4$, whereas $P(A = 0) = 1/2$, $P(B = 1, C = 0) = 1/4$ Thus, $P(A = 0, B = 1, C = 0) \neq P(A = 0) \times P(B = 1, C = 0)$. So A is not independent of (B,C). ■

Bases on Theorem 1 and Lemma 1, we infer the importance of independence among features as expressed in Lemma 2.

Lemma 2: We can not use Theorem 1 unless all the features in the given set are mutually independent. Otherwise, it is unreliable because of Lemma 1.

In practice, in many real-world high-dimensional datasets, only a few features are correlated, most of them are independent. Therefore, we can safely rely on Theorem 1 to compute the joint entropy for a feature set comprising mutually independent features. When the feature sets include part of the dependent features, theorem 1 can not be used directly. However, it can still be used to estimate the bounds for the joint entropies of the feature sets, such that some unpromising candidates can be filtered out first. Next, for the remaining candidate sets, we still need to compute the exact joint entropy values. Therefore, we introduce Lemma 3 in *IIS*.

Lemma 3: Let fs be a subset of size k from *MaxSet*. fs can never become the most informative featureset, if its maximum entropy is lower than *BRFSet*.

Proof: Suppose we have two candidates, *fs* and *BRFSet*. Because the features in *BRFSet* are top-K mutually independent features, the joint entropy of *BRFSet* will be the sum entropy of the individual features according to theorem 1. As fs's entropy is lower than that of *BRFSet*, we will choose *BRFSet* as the most informative featureset. ■

Given the above lemmas and discussions, we finally provide the following Heuristic Feature Reduction Theory: **the most informative feature set is more likely to be a subset from MaxSet**.

TABLE II
THE JOINT DISTRIBUTION (LEFT) AND PROBABILITY (RIGHT) OF A, B AND C

A	B	C
0	1	0
1	0	0
0	0	1
1	1	1

A	**(B,C)**			
	(0,0)	**(0,1)**	**(1,0)**	**(1,1)**
0	P(A=0,B=0,C=0) = 0	P(A=0,B=0,C=1) = 1/4	P(A=0,B=1,C=0) = 1/4	P(A=0,B=1,C=1) = 0
1	P(A=1,B=0,C=0) = 1/4	P(A=1,B=0,C=1) = 0	P(A=1,B=1,C=0) = 0	P(A=1,B=1,C=1)= 1/4
	P(B=0,C=0) = 1/4	P(B=0,C=1) = 1/4	P(B=1,C=0) = 1/4	P(B=1,C=1) = 1/4

This heuristic theory can be illustrated in three folds. In the first place, the most informative feature set from the independent set is *BRFSet*. *BRFSet* is the last and reference candidate we have. We can see from lemma 3 that if the upper bound of a featureset's entropy is smaller than the entropy of *BRFSet*, it can be safely discarded. Next, features in $\{MaxSet - BRFSet\}$ already contain all the top-ranked dependent features and their correlated ones, from which we can select a subset of most informative featuresets. Finally, now that each feature in *BRFSet* is independent of any other feature in *MaxSet*, it is more likely that the one from *BRFSet*, instead of the remaining independent features, will contribute more to the joint entropy of a featureset. In subsection IV-E, we will build the *IIS* framework based on this theory.

B. MIFS: Mining Highly Informative Featureset

With the above heuristic feature reduction theory, we can greatly reduce the number of features for highly informative featureset discovery. Even so, we may still have a very large number of candidates. Let d denote the number of features, n is the number of transactions and k is the size of features set, the number of candidate sets having size k is:

$$nc = \frac{d!}{k! \times (d-k)!} \tag{7}$$

For instance, if $d = 40$, $k = 20$, then $nc = 1.3785 \times 10^{11}$. Therefore we can not afford to enumerate all the candidates of size k, instead, we have to resort to the heuristic algorithms for the consideration of efficiency.

Our proposed feature selection algorithm is named *MIFS*. Initially, the top two features are the pair features with the largest joint entropy. We insert them into a set *mifs*. Then iteratively, we first check the first feature f in $\{MaxSet - MIFS\}$. If f and features in *mifs* are mutually independent, then we simply add f to *mifs*; otherwise, based on the current *mifs* and f, we generate all the possible candidates having size i. At the end of this round, for all the candidates, we scan the data to compute their exact entropy values and find a feature subset *Maxsubset* (of size i) with the largest entropy value. In the next iteration $i+1$, we vacuum the candidate features in the iteration i. i gradually increases from 2 to k, and after $k-2$ iterations, we will have the final highly informative feature set *mifs* of size k.

C. Selecting Only the Promising Featuresets

In this part we present a strategy which can help us filter out the unpromising featuresets at each iteration in the above *MIFS* algorithm, based on Lemma 3. We compare the bounds of each candidate feature set with the bounds the reference. Initially, the reference set is the top-i features in *BRFSet*, i is the current iteration number in algorithm *MIFS*. Let *lb* and *ub* denote the lower and upper bounds. There are 6 cases we need to consider. Bounds comparison is simple so we skip the comparison details. The final strategy is: candidates whose entropy upper bounds are lower than lb will be discarded without consideration; all existing candidates will be removed when there is a featureset whose entropy lower bound is larger than *ub*, and this featureset will be considered as the new reference; in the rest cases we update *lb* or *ub* correspondingly.

D. MFI Data Structure: Pattern Mapping and Feature Indexing for feature selection

The complexity of the *MIFS* algorithm is $|k| * |m| * |n|$, where k is the size of the feature set, $|m|$ is the total number of features in *MaxSet*, $|n|$ is the size of the data. We find that we have to scan the data several times, which is costly. In order to reduce the cost, we make a data structure *MFI* to help us efficiently find the related records. We maintain two kinds of data structures in *MFI*:

(i) A pattern map/table which summarizes all the appeared feature sets and their counts, if two users share the same feature sets, then the count for the pattern in the map is 2;

(ii) Feature indices which keep all the *ids* of the patterns containing this feature. Note that, as we give an order of the patterns by their first occurrence time, the pattern *ids* in each feature index are monotone increasing. This monotonic feature facilitates the search and merge of *ids* from different indices.

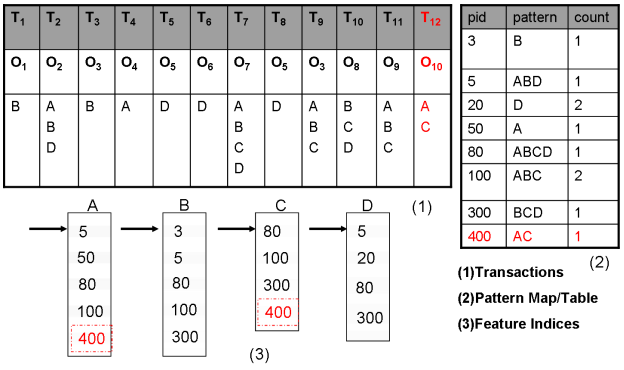

Fig. 1. Indexing the High-dimensional Data

Example 2: Data management in *MFI*. In Figure 1 (1) we have the transactions in the data, and the data structure for

TABLE III
INFORMATION ABOUT THE DATASETS

Dataset	transactions	features	users
dataset1	120000	224	28581
dataset2	240000	264	51611
dataset3	300000	285	65361
dataset4	120000	1351	36423
dataset5	240000	1662	67470
dataset6	300000	1849	74538

storing and identifying the complete patterns is given in Figure 1 (2). Figure 1 (3) is the indices for different features of A, B, C and D. Each time we read a transaction, we check whether the feature set of the object already exists in the pattern map. If yes, we increase the count for the pattern by 1. Otherwise, we first insert the feature set in the pattern map with *pid* as the identifier, then add the new *pid* to the end of related indices. For instance, for the red color transaction T_{12}, because the pattern (A, C) does not exist in the pattern map, we insert the pattern. Next we insert the according pattern id in indices for features A and C.

When we are incrementally searching for the most informative feature set, at each step, we need to compute the entropy of all the new feature subsets which are generated by adding one feature to the current most informative feature subset. There we need *MFI* to quickly find the relevant records.

Example 3: Efficient Computations of Entropy in *MFI*. Let (A, B) be the most informative feature subset in the current round. In the following round, we will have to compute the entropy of (A, B, C) and the combination of (A, B) with either of the remaining features. Owing to our index, when computing the entropy of featureset (A, B, C), we do not have to navigate all the records, instead, all we need to check are patterns kept in index A, B, or C. Thus the computation cost on the entropy will be greatly cut.

According to Formula 7, the search space rapidly increases with the number of features. Thus, the larger the data and the number of features, the more useful *MFI* is.

E. The IIS Framework

We build the *IIS* framework that implements our informative feature sets discovery theory proposed in sub-session IV-A, generates featureset candidates from *MaxSet* using *MIFS* described in sub-session IV-B, then removes unpromising candidates using techniques in subsection IV-C. When computing the exact entropies of the remaining techniques, it utilizes techniques in sub-session IV-D. *MIFS* stops when the length of the selected feature set is k.

V. EXPERIMENTS

Data Set. We use two kinds of real-world datasets in the experiments: Web logs kept by the server at our research institute, and data from a Telecom that maintains the portal visits made by the clients using their mobile devices. The information of the datasets are depicted in Table III in which the first 3 datasets belongs to the Web log dataset and the rest are part of Telecom dataset . There the field of users denotes the number of unique users appeared in the dataset, but one user could request different pages at separate transactions, so we summarize the transactions by users.

Parameter Setting. There are two thresholds for relevance and redundancy, and one user-specified parameter: k, which is the size/length of the feature set. Regarding the thresholds, ξ is set to 0.01, and the value for δ is 0.99. We tested different parameters, and these two values are empirical ones.

We use *MIKI* from [4] as the reference for evaluating the informative feature set mining problem. For efficiency, we use CPU time as the criteria. For evaluating the quality of the algorithms, we use the selected feature sets and their entropy values as the standards. We first look at the selected features, if they are the same, then both algorithms are considered having the same effectiveness. Otherwise, we will check the according entropies, and the one with higher entropy is deemed superior.

Results of *WWW* datasets. (A), (B) and (C) in table IV show the experimental results on dataset1, dataset2 and dataset3. In table IV (A), the "selected features" field denotes the selected informative feature sets, given different k values (because the names of the features are long URL addresses, we only unique *ids* to represent them). The terms such as "+17" mean that the new selected features (e.g. $k = 6$) are composed of feature 17 and the previous result (e.g. $k = 5$). We have the following observations from the results.

(I)No matter how the value of k varies, the selected features and their respective entropy of *IIS* is always the same as *MIKI*. This observation verified our heuristic theory. For example, when k is 10, the feature sets discovered by *MIKI* and *IIS* are the same, and so it does with the according entropy values.

(II)No matter how the value of k varies, *IIS* is always much more efficient than *MIKI*. We can see that for *WWW* datasets, *IIS* saves up to 90% of the time cost by *MIKI*. As an example, when k is 15, *MIKI* takes 898 seconds to get the result, but *IIS* only uses 90 seconds.

(III) For the same dataset, when k increases, it takes much more time to have the final results. *MIKI* suffers a lot when k increases, but the case is much more better for *IIS*.

The above observations still hold for dataset2 and dataset3. We also observe that when k is the same, the more features a dataset has, the longer time it takes. Because the more features we have, the more candidates need to be evaluated. But since *IIS* limits the candidate features to a small subset, it suffers less than *MIKI*. For instance, for the results when k is 30 on dataset2 and dataset3, the computation time for *MIKI* increased by 38.2%, but *IIS* only increased 28.2%.

Results of *Telecom* datasets. We also tested *IIS* on *Telecom* datasets, which have much more features than *WWW* datasets. (D), (E), and (F) in table IV show the results of both algorithms on dataset4, dataset5 and dataset6. Similar to the results of *WWW* datasets, the selected features and entropy are always the same for both algorithms, but *IIS* uses less time. For example, when $k = 5$, *MIKI* uses 3641 seconds on testdata5, but our approach only uses 52 seconds, which is 1.4% of that for *MIKI*. It is clear that when the number of features is very large, *IIS* is much more efficient.

TABLE IV
RESULTS OF *WWW* (A,B,C) AND *Telecom* (D,E,F) DATASETS

(A) DATASET1

K	MIKI			IIS		
	time(s)	selected features	entropy	time(s)	selected features	entropy
5	141	2 4 11 18 20	0.4866	12	2 4 11 18 20	0.4866
6	192	+17	0.5588	16	+17	0.5588
7	247	+25	0.6284	21	+25	0.6284
8	307	+13	0.6954	27	+13	0.6954
9	377	+51	0.7582	33	+51	0.7582
10	448	+24	0.8164	41	+24	0.8164
15	898	+1 15 21 31 37	1.0560	90	+1 15 21 31 37	1.0560
20	1502	+27 29 30 44 64	1.2303	164	+27 29 30 44 64	1.2303
25	2255	+7 8 23 41 54	1.3576	266	+7 8 23 41 54	1.3576
30	3094	+10 19 39 42 50	1.4510	398	+10 19 39 42 50	1.4510

(B) DATASET2

K	MIKI		IIS	
	time(s)	entropy value	Time	entropy value
5	301	0.44826	23	0.44826
6	408	0.516571	34	0.516571
7	524	0.580686	46	0.580686
8	655	0.640754	60	0.640754
9	803	0.699638	75	0.699638
10	957	0.753519	92	0.753519
15	1934	0.9719	209	0.9719
20	3271	1.13181	380	1.13181
25	4888	1.24471	615	1.24471
30	6800	1.3296	912	1.3296

(C) DATASET3

K	MIKI		IIS	
	time(s)	entropy value	Time	entropy value
5	410	0.450336	37	0.450336
6	556	0.516729	51	0.516729
7	719	0.582751	67	0.582751
8	894	0.642691	84	0.642691
9	1096	0.701557	105	0.701557
10	1313	0.754748	128	0.754748
15	2648	0.972163	276	0.972163
20	4486	1.13377	497	1.13377
25	6748	1.2476	792	1.2476
30	9403	1.33378	1169	1.33378

(D) DATASET4

K	MIKI			IIS		
	time(s)	selected features	entropy	time(s)	selected features	entropy
5	1560	14 17 24 28 60	0.8596	31	14 17 24 28 60	0.8596
6	2108	+21	0.9284	42	+21	0.9284
7	2716	+15	0.9943	55	+15	0.9943
8	3367	+37	1.0446	69	+37	1.0446
9	4110	+40	1.0849	85	+40	1.0849
10	5187	+20	1.1245	102	+20	1.1245
15	7649	+19 27 36 56 72	1.2890	180	+19 27 36 56 72	1.2890

(E) DATASET5

K	MIKI		IIS	
	time(s)	entropy value	Time	entropy value
5	3641	0.956659	52	0.956659
6	4939	1.01765	71	1.01765
10	9333	1.19006	173	1.19006

(F) DATASET6

K	MIKI		IIS	
	time(s)	entropy value	Time	entropy value
5	4327	0.753494	54	0.753494
6	5859	0.814299	74	0.814299
10	10949	0.981289	177	0.981289

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a heuristic theory to reduce the dimension for informative feature set discovery from high dimensions. We also introduced structures and a strategy to accelerate the computation of exact entropy of the candidates and prune hopeless featuresets. In the future work we will investigate how to adapt our method to high-dimensional streaming data.

REFERENCES

[1] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[2] H. Heikinheimo, E. Hinkkanen, H. Mannila, T. Mielikäinen, and J. K. Seppänen. Finding low-entropy sets and trees from binary data. In *SIGKDD '07*, pages 350–359, New York, NY, USA, 2007. ACM.

[3] J. Jaeger, R. Sengupta, and W. Ruzzo. Improved gene selection for classification of microarrays. In *Pacific Symposium on Biocomputing*, pages 53–64, 2002.

[4] A. J. Knobbe and E. K. Y. Ho. Maximally informative k-itemsets and their efficient discovery. In *SIGKDD '06*, pages 237–244, New York, NY, USA, 2006. ACM.

[5] H. Liu and H. Motoda. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007.

[6] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

[7] J. A. T. Thomas M. Cover. *Elements of Information Theory, 2nd Edition*. Wiley, 2006.

[8] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning (ICML)*, pages 412–420. Morgan Kaufmann Publishers, 1997.

[9] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *International Conference on Machine Learning (ICML)*, pages 856–863, 2003.

[10] C. Zhang and F. Masseglia. Discovering highly informative feature sets from data streams. In *21st International Conference on Database and Expert Systems Applications (DEXA 2010)*, pages 91–104, 2010.

[11] X. Zhang, F. Zou, and W. Wang. Fastanova: an efficient algorithm for genome-wide association study. In *SIGKDD*, pages 821–829, 2008.