TOP-COP: Mining TOP-K Strongly Correlated Pairs in Large Databases

Hui Xiong [*]
MSIS Department, Rutgers University
hui@rbs.rutgers.edu

Mark Brodie
IBM TJ Watson
mbrodie@us.ibm.com

Sheng Ma
Vivido Media Inc.
shengma2005@gmail.com

Abstract

Recently, there has been considerable interest in computing strongly correlated pairs in large databases. Most previous studies require the specification of a minimum correlation threshold to perform the computation. However, it may be difficult for users to provide an appropriate threshold in practice, since different data sets typically have different characteristics. To this end, we propose an alternative task: mining the top-k strongly correlated pairs. In this paper, we identify a 2-D monotone property of an upper bound of Pearson's correlation coefficient and develop an efficient algorithm, called TOP-COP to exploit this property to effectively prune many pairs even without computing their correlation coefficients. Our experimental results show that the TOP-COP algorithm can be orders of magnitude faster than brute-force alternatives for mining the top-k strongly correlated pairs.

1 Introduction

Given a large set of items and observation data about co-occurring items, association analysis is concerned with the identification of strongly related subsets of items. Association analysis is a core problem in data mining and databases. It plays an important role in many application domains such as market-basket analysis [2], climate studies [14], public health [5], and bioinformatics [17].

The focus of this paper is on computing a *top-k correlated-pairs query* that returns the top k pairs of positively correlated items. As a motivating example, the top-k correlated-pairs query can reveal information about how the sales of a product are related to the sales of other products. This type of information can be useful for sales promotions, catalog design, and store layout. However, as the number of items and transactions in the data set increases, the computational cost of the top-k correlated-pairs query becomes prohibitively expensive. For example, if a database contains 10^6 items, which may represent the collection of books available at an e-commerce Web site, a brute-force approach to answering the top-k correlated-pairs query requires computing the correlations of $\binom{10^6}{2} \approx 0.5 \times 10^{12}$ possible item pairs. Thus, it may not be computationally feasible to apply a brute-force approach.

The *top-k correlated-pairs query* problem can be described as follows: Given a user-specified k and a database with N items and T transactions, a top-k correlated-pairs query finds the top k item pairs with the highest positive correlations. The scope of the top-k correlated-pairs query problem proposed in this paper is restricted to market basket databases with binary variables, and the form of correlation is Pearson's correlation coefficient [13] for binary variables, also called the ϕ correlation coefficient.

The main contribution of this work is the development of the **TOP**-k **CO**rrelated-**P**airs (TOP-COP) query algorithm. We show that TOP-COP finds the top-k correlated pairs in a computation time which can be orders of magnitude faster than the brute-force alternative. The algorithm exploits a 2-D monotone property of the upper bound of Pearson's correlation coefficient. By interpreting this property geometrically, we obtain an algorithm that uses a diagonal traversal method, combined with a refine-and-filter strategy, to efficiently find the top-k pairs.

Related Work. The top-k correlated-pairs query problem is different from the standard association-rule mining problem [1, 3, 12, 15, 16]. Given a set of transactions, the objective of association rule mining is to extract all subsets of items that satisfy a minimum support threshold. Support measures the fraction of transactions that contain a particular subset of items.

The notions of support and correlation may not necessarily agree with each other. This is because item pairs with high support may be poorly correlated while those that are highly correlated may have low support. For instance, suppose we have an item pair {A, B}, where $supp(A) = supp(B) = 0.8$ and $supp(A, B) = 0.64$. Both items are uncorrelated, since $supp(A, B) = supp(A)supp(B)$. In contrast, an item pair {A, B} with $supp(A) = supp(B) = supp(A, B) = 0.001$ is perfectly correlated despite its low support. Patterns with low support but high correlation

[*] Contact Author.

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

are useful for capturing interesting associations among rare anomalous events or rare but expensive items such as gold necklaces and earrings.

Recently, Xiong et al. [18] proposed the TAPER algorithm to efficiently compute the all-strong-pairs correlation query. Given a user-specified minimum correlation threshold θ and a database with N items and T transactions, the all-strong-pairs correlation query finds all item pairs with correlation coefficients above the threshold θ. However, it is difficult for users to provide an appropriate correlation threshold for the all-strong-pairs correlation query, since different data sets typically have different characteristics.

Along the same line of the all-strong-pairs correlation query, Ilyas et al. [7] also proposed a method for efficiently identifying correlated pairs. In this method, sampling techniques are applied to exploit efficient computation. As a result, this method cannot avoid false-positive and false-negative correlations. Furthermore, this method also requires users to specify a correlation threshold.

Additionally, Jermaine [8] investigated the implication of incorporating chi-square (χ^2) [13] based queries to data cube computations. He showed that finding the subcubes that satisfy statistical tests such as χ^2 are inherently NP-hard, but can be made more tractable using approximation schemes. Jermaine [9] also presented an iterative procedure for high-dimensional correlation analysis by shaving off part of the database via feedback from human experts. Finally, Brin [4] proposed a χ^2-based correlation mining strategy; however χ^2 does not possess an upward closure property for exploiting efficient computation [6].

2 Basic Concepts

In statistics, a measure of association is a numerical index which describes the strength or magnitude of a relationship among variables. Although literally dozens of measures exist, they can be categorized into two broad groups: ordinal and nominal. Relationships among ordinal variables can be analyzed with ordinal measures of association such as Kendall's Tau [10] and Spearman's Rank Correlation Coefficient [11]. In contrast, relationships among nominal variables can be analyzed with measures of association such as Pearson's Correlation Coefficient and measures based on Chi Square [13].

The ϕ correlation coefficient [13] is the computational form of Pearson's Correlation Coefficient for binary variables. In this section, we describe some basic concepts related to the ϕ correlation coefficient.

In a 2 × 2 two-way table shown in Table 1, the calculation of the ϕ correlation coefficient reduces to

$$\phi_{\{A,B\}} = \frac{P_{(00)}P_{(11)} - P_{(01)}P_{(10)}}{\sqrt{P_{(0+)}P_{(1+)}P_{(+0)}P_{(+1)}}} \quad (1)$$

		B		Row Total
		0	1	
A	0	$P_{(00)}$	$P_{(01)}$	$P_{(0+)}$
	1	$P_{(10)}$	$P_{(11)}$	$P_{(1+)}$
Column Total		$P_{(+0)}$	$P_{(+1)}$	N

Table 1. A two-way table of item A and item B.

where $P_{(ij)}$, for i = 0, 1 and j = 0, 1, denote the number of samples which are classified in the ith row and jth column of the table. Furthermore, we let $P_{(i+)}$ denote the total number of samples classified in the ith row and $P_{(+j)}$ denote the total number of samples classified in the jth column. Thus $P_{(i+)} = \sum_{j=0}^{1} P_{(ij)}\ and\ P_{(+j)} = \sum_{i=0}^{1} P_{(ij)}$.

When adopting the support measure of association rule mining [1], for two items A and B in a market basket database, we have $supp(A) = P_{(1+)}/N$, $supp(B) = P_{(+1)}/N$, and $supp(A,B) = P_{(11)}/N$, where N is the total number of samples in the two-way table. With support notations, as illustrated in [18], we have the support form of Equation 1 shown below as Equation 2.

$$\phi_{\{A,B\}} = \frac{supp(A,B) - supp(A)supp(B)}{\sqrt{supp(A)supp(B)(1-supp(A))(1-supp(B))}} \quad (2)$$

(a) A Market Basket Database

TID	Items
1	a, b, c
2	a, b, c
3	a, c
4	a, b
5	a, b
6	a, b
7	a, b, c, d, e, f
8	a, b, d, e
9	a, b, d
10	c

(b) Item Pairs with Upper Bounds and Correlation Coefficients

Pair	Upper Bound	Correlation
{a, b}	0.667	0.667
{a, c}	0.333	−0.333
{a, d}	0.218	0.218
{a, e}	0.167	0.167
{a, f}	0.111	0.111
{b, c}	0.5	−0.5
{b, d}	0.327	0.327
{b, e}	0.25	0.25
{b, f}	0.167	0.167
{c, d}	0.655	−0.218
{c, e}	0.5	0
{c, f}	0.333	0.333
{d, e}	0.764	0.764
{d, f}	0.509	0.509
{e, f}	0.667	0.667

Figure 1. An Example Data Set.

For example, consider the example data set shown in Figure 1 (a). To compute $\phi_{\{a,b\}}$, we note that $supp(a) = 9/10$, $supp(b) = 8/10$, and $supp(a,b) = 8/10$. Direct calculation shows that $\phi_{\{a,b\}} = 0.08/0.12 = 2/3$, confirming that a and b are strongly correlated.

Given an item pair $\{A, B\}$, the support $supp(A)$ for item A, and the support $supp(B)$ for item B, we can suppose without loss of generality that $supp(A) \geq supp(B)$. Xiong et al. [18] derived an upper bound $upper(\phi_{\{A,B\}})$ of the ϕ correlation coefficient for $\{A,\ B\}$ as follows:

$$\phi_{\{A,B\}} \leq upper(\phi_{\{A,B\}}) = \sqrt{\frac{supp(B)}{supp(A)}}\sqrt{\frac{1-supp(A)}{1-supp\ (B)}} \tag{3}$$

3 A 2-D Monotone Property of the Upper Bound of ϕ Correlation Coefficient

In this section, we present a 2-D monotone property of the upper bound of the ϕ correlation coefficient. This monotone property can be exploited to develop a diagonal traversal method for efficiently mining the top-k correlated pairs.

Using Equation 3 above, Xiong et al. [18] also derived a 1-D monotone property of the upper bound of the ϕ correlation coefficient:

Lemma 1 *For an item pair* $\{A,\ B\}$*, let* $supp(A) > supp(B)$ *and fix item A. The upper bound,* $upper(\phi_{\{A,B\}})$*, is monotone decreasing with decreasing support of item B.*

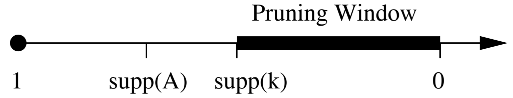


Figure 2. A Geometric Interpretation of the Computational Exploration of the 1-D Monotone Property

Figure 2 shows a geometric interpretation of the computational exploration of the 1-D monotone property of the upper bound. In the figure, all items are sorted according to support values in non-increasing order. Let us consider the all-strong-pairs correlation query with a user-specified correlation threshold θ. If we identify an item k such that the upper bound $upper(\phi_{\{A,k\}})$ is less than the threshold θ, then any pair {A, f} with $supp(f) < supp(k)$ can be safely pruned, since the upper bound $upper(\phi_{\{A,f\}})$ is guaranteed to be less than $upper(\phi_{\{A,k\}})$ according to Lemma 1. In other words, for any item A, we can generate a one-dimensional pruning window for efficiently eliminating pairs which do not satisfy the correlation threshold θ.

Lemma 2 *For a pair of items* $\{A,\ B\}$*, let* $supp(A) > supp(B)$ *and fix item B. The upper bound* $upper(\phi_{\{A,B\}})$ *is monotone increasing with decreasing support of item A.*

Proof: Given two items A_1 and A_2 with $supp(A_1){>}supp(A_2) > supp(B)$, we only need to prove that $upper(\phi_{\{A_2,B\}}) > upper(\phi_{\{A_1,B\}})$. According to Equation 3, $upper(\phi_{\{A,B\}}) = \sqrt{\frac{supp(B)}{supp(A)}}\sqrt{\frac{1-supp(A)}{1-supp(B)}}$, with $A = A_1$ and $A = A_2$, we see that $\frac{upper(\phi_{\{A_2,B\}})}{upper(\phi_{\{A_1,B\}})} = \sqrt{\frac{supp(A_1)}{supp(A_2)}}\sqrt{\frac{1-supp(A_2)}{1-supp(A_1)}} > 1$ because $supp(A_1) > supp(A_2)$ and $(1 - supp(A_1)) < (1 - supp(A_2))$.

Lemma 1 and Lemma 2 form the basis of the 2-D monotone property of the upper bound, illustrated in Figure 3. An item list $\{a, b, c, d, e, f\}$, is sorted by item support values in non-increasing order. The upper bound of item pairs decreases following the direction of the arrow. For instance, the upper bound of item pair $\{d, f\}$ is greater than that of item pair $\{c, f\}$ but smaller than that of item pair $\{d, e\}$.

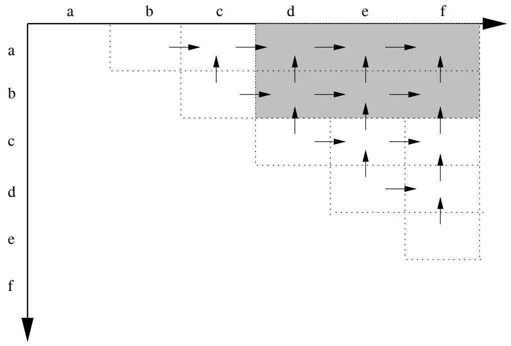


Figure 3. A Geometric Interpretation of the Computational Exploration of the 2-D Monotone Property

In contrast to Figure 2, the 2-D monotone property can help prune item pairs from two dimensions instead of one dimension. For instance, if the upper bound of item pair $\{b, d\}$ indicates that this pair is not strongly correlated, we can generate a rectangle pruning space for efficiently eliminating item pairs: $\{b, e, \}$, $\{b, f\}$, $\{a, d\}$, $\{a, e\}$, and $\{a, f\}$. Since the upper bounds of all these five item pairs cannot be greater than the upper bound of item pair $\{b, d\}$, these pairs are also not strongly correlated.

4 TOP-COP: TOP-K COrrelated Pairs Query

Here, we introduce the **TOP**-k **CO**rrelated **P**airs (TOP-COP) query algorithm. The key idea of TOP-COP is a diagonal traversal method which exploits the 2-D monotone property of the upper bound of the ϕ correlation coefficient for effciently computing the top-k correlated pairs.

Indeed, a brute-force approach to computing the top-k correlated pairs is to first compute the correlation coefficients for all item pairs, then sort all item pairs based on their correlation coefficients, and finally report the top-k strongly correlated pairs as the final result. In contrast to this brute-force approach, we propose a diagonal-traversal method to efficiently computing top-k correlation pairs.

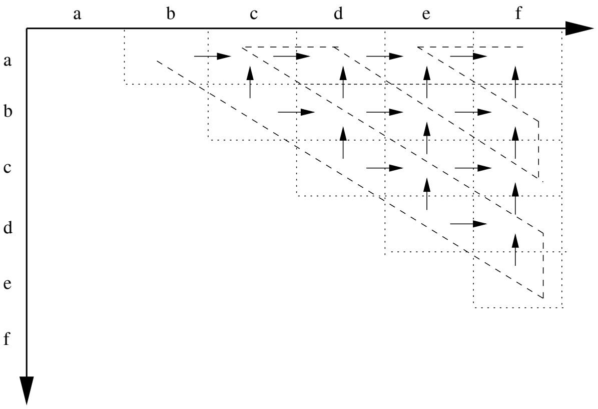
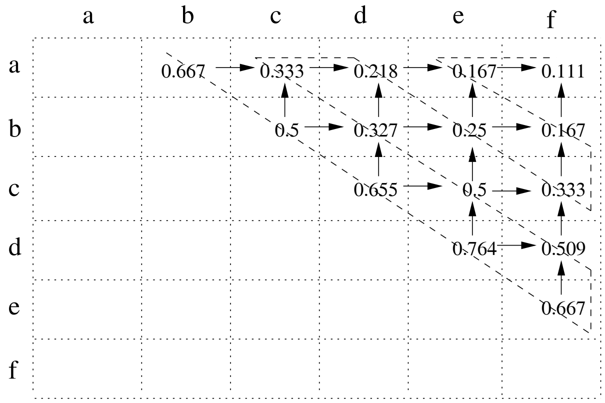


Figure 4. Illustration of Diagonal Traversal.

Figure 4 illustrates the diagonal traversal method. In the figure, an item list $\{a, b, c, d, e, f\}$ is sorted by item support values in non-increasing order. The upper bound of item pairs decreases following the arrow direction according to Lemma 1 and Lemma 2. The diagonal traversal method conducts a diagonal traverse to search for the top-k correlated pairs. The search starts from the principal diagonal which is traversed in the "southeast" direction, then goes to the diagonal above the main diagonal, and so on. During the iterative search process, this method maintains a top-k list and an item pair is sorted-inserted into this list if the correlation coefficient of this item pair is greater than the minimum correlation coefficient in the top-k list. The search stops if the maximal upper bound of all item pairs in a diagonal is less than the current minimum correlation coefficient in the top-k list.



	a	b	c	d	e	f
a		0.667	0.333	0.218	0.167	0.111
b			0.5	0.327	0.25	0.167
c				0.655	0.5	0.333
d					0.764	0.509
e						0.667
f						

Figure 5. An Example of Diagonal Traversal.

Example 1 *To illustrate the diagonal traversal method, consider the data set shown in Figure 1 (a). To simplify the discussion, we use an item list $\{a, b, c, d, e, f\}$ which is sorted by item support values in non-increasing order. Figure 1 (b) shows the upper bound and correlation coefficients for every item pair. We can arrange all item pairs in the upper triangle of a matrix as shown in Figure 5. Suppose that we are interested in finding the top-3 strongly correlated item pairs from the data set. After traversing the main diagonal, we have a top-3 list containing three item pairs $\{e, f\}$, $\{d, e\}$, and $\{a, b\}$. The minimum correlation coefficient in this top-3 list is 0.667. Next, we search the super-diagonal. We find that the maximal upper bound of all item pairs in the super-diagonal is 0.509 and is less than 0.667. Therefore, the search stops.*

5 Experimental Results

Our experiments were performed on two real-life data sets, described further below. All experiments were performed using code in C++ on a Sun Ultra 10 workstation with a 440 MHz CPU and 128 Mbytes of memory running the SunOS 5.7 operating system.

Data set	#Item	#Transaction	Source
Pumsb	2113	49046	IBM Almaden
LA1	29704	3204	TREC-5

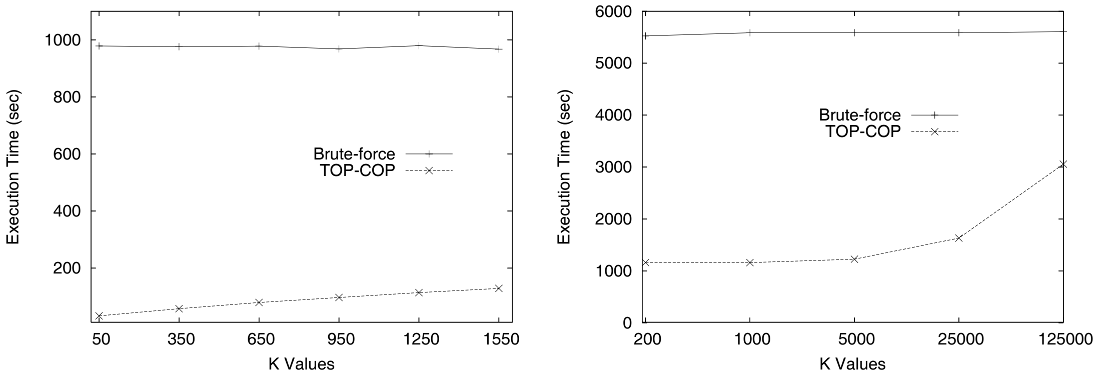
Table 2. Real-life Data Set Characteristics.

Real-life Data Sets. The real-life data sets were obtained from several different application domains. Table 5 shows some characteristics of these data sets. `Pumsb` is often used as the benchmark for evaluating the performance of association rule algorithms on dense data sets. The `pumsb` data set correspond to binarized versions of a census data set from IBM[1]. In addition, the `LA1` data set is part of the TREC-5 collection (http://trec.nist.gov) and contains news articles from the Los Angeles Times.

A Performance Comparison We present a performance comparison between the TOP-COP algorithm and a brute-force approach using several benchmark data sets from IBM and TREC. The implementation of the brute-force approach is similar to that of the TOP-COP algorithm except that the filtering mechanism implemented in the TOP-COP algorithm is not included in the brute-force approach. Please note that we do not use TAPER [18] as the benchmark, since TAPER is an algorithm for finding all item pairs with correlation above a given correlation threshold and the purpose in this work is to find the top-k correlated item pairs. Also, it is nontrivial to modify the TAPER algorithm for finding the top-k correlated item pairs, because it is difficult to specify the appropriate correlation thresholds.

Figure 6 shows the relative computation performance of the TOP-COP algorithm and the brute-force approach on the `Pumsb` and `LA1` data sets. As can be seen, the performance of the brute-force approach does not change much

[1] These data sets were obtained from IBM Almaden at http://www.almaden.ibm.com/cs/quest/demos.html



(a) Pumsb

(b) LA1

Figure 6. A Performance Comparison.

for any of the two data sets at different top-k values. However, the execution time of the TOP-COP algorithm can be an order of magnitude faster than the brute-force approach at if k is small. For instance, as shown in Figure 6 (a), the execution time of the TOP-COP algorithm on the `Pumsb` data set is one order of magnitude less than that of the brute-force approach when the K values are less than 950. Also, as the value of k increases, the execution time of the TOP-COP algorithm increases accordingly. Similar computation effects can also be observed on the `LA1` data set, although the computation savings on this data set is not as significant as on the $Pumsb$ data set.

6 Conclusions

In this paper, we identify a 2-D monotone property of the upper bound of Pearson's correlation coefficient and provide a geometric interpretation of the computational exploration of the 2-D monotone property for efficiently computing the top-k correlated pairs. With this 2-D monotone property, we designed a **TOP**-k **CO**rrelated **P**airs (TOP-COP) query algorithm that uses a diagonal traversal method, combined with a refine-and-filter strategy, to efficiently find the top-k correlated pairs. In addition, as demonstrated by our experiments, the TOP-COP algorithm can be orders of magnitude faster than the brute-force alternative for mining the top-k correlated pairs.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD*, pages 207–216, 1993.

[2] C. Alexander. *Market Models: A Guide to Financial Data Analysis*. John Wiley & Sons, 2001.

[3] R. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery Journal*, pages 217–240, 2000.

[4] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD*, pages 265–276, 1997.

[5] P. Cohen, J. Cohen, S. G. West, and L. S. Aiken. *Applied Multiple Regression/Correlation Analysis for the Behavioral Science*. Lawrence Erlbaum Assoc; 3rd edition, 2002.

[6] W. DuMouchel and D. Pregibon. Empirical bayes screening for multi-item associations. In *KDD*, pages 67–76, 2001.

[7] I. F. Ilyas, V. Markl, P. J. Haas, P. Brown, and A. Aboulnaga. Cords: Automatic discovery of correlations and soft functional dependencies. In *ACM SIGMOD*, 2004.

[8] C. Jermaine. The computational complexity of high-dimensional correlation search. In *ICDM*, 2001.

[9] C. Jermaine. Playing hide-and-seek with correlations. In *ACM SIGKDD*, 2003.

[10] M. Kendall and J. D. Gibbons. *Rank Correlation Methods*. Oxford University Press (5th edition), 1990.

[11] E. L. Lehmann and H. j. M. D'Abrera. *Nonparametrics: Statistical Methods Based on Ranks*. Prentice Hall, 1998.

[12] B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. In *ACM SIGKDD*, 1999.

[13] H. T. Reynolds. *The Analysis of Cross-classifications*. The Free Press, New York, 1977.

[14] H. V. Storch and F. W. Zwiers. *Statistical Analysis in Climate Research*. Cambridge University Press, 2002.

[15] K. Wang, Y. He, D. Cheung, and Y. Chin. Mining confident rules without support requirement. In *ACM CIKM*, 2001.

[16] H. Xiong, , S. Shekhar, P. Tan, and V. Kumar. Taper: An efficient two-step approach for all-pairs correlation query in transaction databases. In *Technical Report 03-020, computer science and engineering, University of Minnesota - Twin Cities*, May 2003.

[17] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. In *Proc. of the Pacific Symposium on Biocomputing (PSB)*, January 2005.

[18] H. Xiong, S. Shekhar, P. Tan, and V. Kumar. Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In *ACM SIGKDD*, pages 334–343, 2004.

0-7695-2701-9/06 $20.00 © 2006 **IEEE**