Greedy Localization, Iterative Compression and Modeled Crown Reductions: New FPT Techniques, an Improved Algorithm for Set Splitting and a Novel $2k$ Kernelization for Vertex Cover

Frank Dehne[1], Mike Fellows[2], Frances Rosamond[2] and Peter Shaw[2]

[1] Griffith University, Brisbane QLD 4111, Australia
frank@dehne.net
[2] University of Newcastle, Callaghan NSW 2308, Australia
{mfellows,fran,pshaw}@cs.newcastle.edu.au

Abstract. The two objectives of this paper are: (1) to articulate three new general techniques for designing FPT algorithms, and (2) to apply these to obtain new FPT algorithms for Set Splitting and Vertex Cover. In the case of Set Splitting, we improve the best previous $\mathcal{O}^*(72^k)$ FPT algorithm due to Dehne, Fellows and Rosamond [DFR03], to $\mathcal{O}^*(8^k)$ by an approach based on *greedy localization* in conjunction with *modeled crown reduction*. In the case of Vertex Cover, we describe a new approach to $2k$ kernelization based on *iterative compression* and *crown reduction*, providing a potentially useful alternative to the Nemhauser-Trotter $2k$ kernelization.

1 Introduction

This paper has a dual focus on: (1) the exposition of some new general FPT algorithm design techniques, and (2) the description of two concrete applications of these techniques to the Vertex Cover and Set Splitting problems. The latter is defined:

Set Splitting
Instance: A collection $\mathcal{F}$ of subsets of a finite set X, and a positive integer k.
Parameter: k
Question: Is there a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ and a partition of X into disjoint subsets X_0 and X_1 such that for every $S \in \mathcal{F}', S \cap X_0 \neq \varnothing$ and $S \cap X_1 \neq \varnothing$, with $|\mathcal{F}'| \geq k$?

The Set Splitting problem is NP-complete [GJ79] and APX-complete [Pe94]. Andersson and Engebretsen [AE97], and Zhang and Ling [ZL01] presented approximation algorithms that provide solutions within a factor of 0.7240 and 0.7499, respectively. A 1/2 approximation algorithm for the version of the Set Splitting problem where the size of X_0 is specified has been described by Ageev and Sviridenko [AS00].

It is a straightforward exercise to show that Set Splitting is fixed parameter tractable by the method of color-coding [AYZ95]. One of the techniques we will discuss, which we refer to here as *greedy localization* was first used by Chen, Friesen, Jia and Kanj [CFJK01] (see also Jia, Zhang and Chen [JZC03]). This approach can potentially be employed in designing FPT algorithms for many different *maximization* problems. In the case of Set Splitting we present an example of the deployment of this approach that yields a significant improvement over the best previous $\mathcal{O}^*(72^k)$ FPT algorithm for this problem, due to Dehne, Fellows and Rosamond [DFR03]. Here we combine this technique with *crown reduction* (where the reduction rule is guided by a crown decomposition of an associated graph that models the situation) and obtain an $\mathcal{O}^*(8^k)$ FPT algorithm for Set Splitting.

The method of *iterative compression* could be viewed as in some sense "dual" to greedy localization, since it seems to be potentially applicable to a wide range of *minimization* problems. Both of these techniques are in the way of "opening moves" that can be used to develop some initial structure to work with. Neither is technically deep, but still they can be pointed to as potentially of broad utility in FPT algorithm design. A simple application of iterative compression to Vertex Cover yields a new $2k$ Turing kernelization that may offer practical advantages over the Nemhauser-Trotter $2k$ many:1 kernelization algorithm.

We assume that the reader has a basic familiarity with the fundamental concepts and techniques in the FPT toolkit, as exposited in [DF99,Nie02] (and also with the definition of basic combinatorial problems such as Vertex Cover). We also assume that the reader is familiar with the research program in "worst-case exponential complexity" articulated in the survey paper by Woeginger [Woe03]. In particular, we employ the handy $\mathcal{O}^*$ notation introduced there for FPT results, that suppresses the polynomial time contribution of the overall input size and focuses attention on the exponential time-complexity contribution of the declared parameter. An FPT algorithm that runs in time $\mathcal{O}^*(8^k)$ thus runs in time $\mathcal{O}(8^k n^c)$ for some constant c independent of the parameter k.

2 The New Techniques

There are three new FPT design techniques to which we wish to draw attention:

- greedy localization
- iterative compression
- modeled crown reductions

The first two are only relatively simple opening moves, but nevertheless these deserve wider recognition in the context of FPT algorithm design.

2.1 Greedy Localization

This is an approach that can often be applied to maximization problems. The idea is to start off with an attempted greedy solution. For example, in the case of the Set Packing algorithm due to Jia, Zhang and Chen, the first step is to greedily compute a maximal collection of pairwise disjoint sets. If k are found, then of course we are done. Otherwise, we can make the observation that if there is any solution (k pairwise disjoint sets) then every set in the solution must intersect our (small) maximal collection. Thus we have gained some initial information that narrows our search, "localizes" our efforts to this initial structure.

As Set Splitting is a maximization problem, we will similarly employ here an opening move that attempts to find a greedy solution, which similarly either succeeds and we are done, or provides us with some initial structure to work with. Greedy localization has been employed in a few other recent FPT algorithms [FHRST04,PS04,MPS04,FKN04].

2.2 Iterative Compression

This "opening move" to develop initial structure seems first to have been used in an FPT algorithm recently described by Reed, Smith and Vetta for the problem of determining for a graph G whether k vertices can be deleted to obtain a bipartite graph G' (an important breakthrough as this problem has been open for some time) [RSV03]. Their approach can be briefly described as follows.

First, the problem is respecified constructively: we aim for an FPT algorithm that either outputs NO, or constructively produces the set of k vertices whose removal will make the graph bipartite.

Second, we attempt a recursive solution (which we will see has a simple iterative interpretation). Choose a vertex v, and call the algorithm on $G - v$. This either returns NO, and we can therefore return NO for G, or it returns a solution set of size k. By adding v to this set, we obtain a solution of size $k + 1$ for G, and what remains to be done is to address the following (constructive) *compression* form of the problem:

Input: G and a solution S of size $k + 1$
Output: Either NO, or a solution of size k, if one exists.

The iterative interpretation is that we are building the graph up, vertex by vertex, and at each step we have a small solution (of size $k + 1$) and attempt to compress it. This interpretation makes clear that our overall running time will be $\mathcal{O}(n \cdot f(n, k))$ where $f(n, k)$ is the running time of our FPT algorithm for the compression form of the problem. Of course, all the real work lies there, but this overall algorithmic approach, simple as it is, gives us some initial structure to work with. The approach is clearly of potential utility for many different minimization problems. (For another recent application see [Ma04].)

2.3 Modeled Crown Reductions

Both of our concrete applications, to the Set Splitting and to the Vertex Cover problems, also use the recently developed techniques of crown decompositions and crown reduction rules. This technique was first introduced by Chor, Fellows and Juedes [CFJ04] (a brief exposition can also be found in the recent survey [F03]). In [CFJ04] the technique is applied to the problems Graph Coloring with $(n-k)$ Colors and to Vertex Cover. Crown reduction has turned out to be effective for a surprisingly wide range of parameterized problems; see also [PS03,FHRST04,PS04,MPS04]. Here we show that crown reductions can even be employed on problems that are not about graphs. Our $\mathcal{O}^*(8^k)$ FPT algorithm for Set Splitting employs a kernelization rule that is based on a crown decomposition in an associated auxiliary graph that models some of the combinatorics of the Set Splitting problem.

The machinery from [CFJ04] that we employ here is next described.

Definition 1. *A* crown decomposition *of a graph $G = (V, E)$ is a partition of the vertices of G into three disjoint sets H, C and J with the following properties:*

1. *C is an independent set in G.*
2. *H separates C from J, that is, there are no edges between C and J.*
3. *H is matched into C, that is, there is an injective assignment $m : H \to C$ such that $\forall h \in H$, h is adjacent to $m(h)$.*

The Crown Rule for Vertex Cover transforms (G, k) into (G', k'), where $G' = G - C - H$, and $k' = k - |H|$. The Crown Rule for the Graph Coloring with $(n-k)$ Colors problem is (surprisingly) the same rule applied to $\bar{G}$ [CFJ04].

We will use the following lemma from [CFJ04].

Lemma 1. *If a graph $G = (V, E)$ has an independent set $I \subseteq V(G)$ such that $|N(I)| < |I|$ then a nontrivial crown decomposition (C, H, J) with $C \subseteq I$ for G can be found in time $\mathcal{O}(|V| + |E|)$.*

3 An $\mathcal{O}^*(8^k)$ FTP algorithm for Set Splitting

The input to the Set Splitting problem consists of a family $\mathcal{F} \subseteq 2^X$ of subsets of a base set X, and a positive integer k. We can trivially assume that every set $S \in \mathcal{F}$ consists of at least two elements of X.

The first step of our algorithm is a greedy computation of what we will term a *witness structure* for the instance. The witness structure consists of a collection of sets $\mathcal{F}' \subseteq \mathcal{F}$, and for each of the sets $S_i \in \mathcal{F}'$ a choice of two distinct elements $b_i \in S_i$ and $w_i \in S_i$. It is allowed that these chosen elements may coincide, that is, for $S_i \neq S_j$ possibly $b_i = b_j$ or $w_i = w_j$ (or both). What is also required of the witness structure is that the sets $B = \{b_1, b_2, ..., b_r\}$ of *black* witness elements, and $W = \{w_1, w_2, ..., w_r\}$ of *white* witness elements are disjoint. It is clear that if we succeed in greedily computing a witness structure with $|\mathcal{F}'| = r$, then any extension of the disjoint subsets B and W of X to a bipartition of X will split the r sets in $\mathcal{F}'$.

The first step (greedy localization) is to compute a *maximal* witness structure by the obvious greedy algorithm of repeatedly adding sets to $\mathcal{F}'$ so long as this is possible. If $r \geq k$ then we are done.

At the end of the greedy step, if we are not done, then the following structural claims hold.

Claim 1. Every set S not in the maximal witness structure collection $\mathcal{F}'$ consists entirely of black or entirely of white elements, that is, either $S \subseteq B$ or $S \subseteq W$.

Proof. We have assumed that every set contains at least two elements. Consider a set $S \in \mathcal{F}$ that is not in the maximal witness structure family $\mathcal{F}'$. If $S \subseteq B \cup W$, then clearly either $S \subseteq B$ or $S \subseteq W$ else S is split and could be added to $\mathcal{F}'$. Hence suppose that there is an element $x \in S$, where $x \notin B \cup W$. If S contains an element of B (or W) then x could be assigned to W (or B) and $\mathcal{F}'$ augmented by S, contradicting our assumption that the witness structure is maximal. Since S has at least two elements, the only remaining case is that S contains two distinct elements that do not belong to $B \cup W$. But then, one could be assigned to B and one to W and $\mathcal{F}'$ could again be augmented, contradicting our assumption.

The following claim is obvious (but crucial):

Claim 2. $|B| \leq k-1$ and $|W| \leq k-1$.

Our algorithm is described as follows:

Step (1): Greedily compute a maximal witness structure. If k sets are split, then report YES and STOP. (If not, then $|\mathcal{F}'| \leq k-1$.)

Step (2): Branch on all ways of "recoloring" the (at most) $2(k-1)$ elements that were colored (placed in either B or W) in the witness structure.

Subproblem
For each recoloring (bipartition) of $B \cup W$ into B' and W'

Step (3): Determine the number of sets that have been split. If k sets are split then report YES and STOP.

Otherwise
Step (4): Generate an auxiliary graph G describing the sets that remain unsplit and the elements of $X - (B \cup W)$ contained in them.

Step (5): Repeatedly apply the Crown Reduction Rule (described below) to the subproblem represented by this graph until a subproblem kernel consisting of at most $(k-1)$ elements not in $B' \cup W'$ remains.

Step (6): After we have exhausted the ability to reduce the subproblem instance using the Crown Reduction Rule, there can be at most $k-1$ vertices still remaining to be assigned a color. Try all 2^{k-1} ways to color these elements.

3.1 The Subproblem

After re-coloring (partitioning) $B \cup W$ into B' and W', some number t of sets in $\mathcal{F}$ will have been split (that is, have nonempty intersection with both B' and W'). If $t \geq k$ then of course we will be done (Step 3). Let $\mathcal{G}$ denote the subfamily of $\mathcal{F}$ that is not split by B' and W'. The subproblem is whether the disjoint sets B' and W' can be extended to a bipartition of X that splits k sets. In other words, the subproblem is to determine if the remaining (yet uncolored) elements, those in $X - (B' \cup W')$ can be colored (black and white, extending B' and W') in such a way that at least $k' = k - t$ further sets in $\mathcal{G}$ are split. Note that the fate of any set that is a subset of $B \cup W$ is completely determined by the recoloring into B' and W': it is either among those split, or no extension can split it. Thus in the subproblem, we can restrict our attention (by Claim 1) to the sets in $\mathcal{G}' = \mathcal{G} - 2^{B \cup W} \subseteq \mathcal{F}'$. That is, the only candidates for further splitting belong to our greedy collection $\mathcal{F}'$ (!) and there are at most $k - 1$ of these. We can therefore observe the following claims concerning the subproblem:

Claim 3. Every set in $\mathcal{G}'$ contains either two distinct elements of B' (denote these sets $\mathcal{B}$) or two distinct elements of W' (denote these sets $\mathcal{W}$). Furthermore, every set in $\mathcal{G}'$ contains at least one element of $X - (B' \cup W')$.

Claim 4. $|\mathcal{B} \cup \mathcal{W}| \leq k - 1$.

3.2 Crown Reduction for the Subproblem

The subproblem is modeled by a bipartite graph with vertex sets $V_{\mathcal{B}} \cup V_{\mathcal{W}}$, and $V_{\mathcal{U}}$. The vertices v_S of $V_{\mathcal{B}} \cup V_{\mathcal{W}}$ correspond, respectively, to the unsplit sets S in $\mathcal{B}$ and $\mathcal{W}$. The vertices u_x of $V_{\mathcal{U}}$ correspond to the uncolored elements in $\mathcal{U} = X - (B' \cup W')$. There is an edge between v_S and u_x if and only if $x \in S$. See Figure 2.

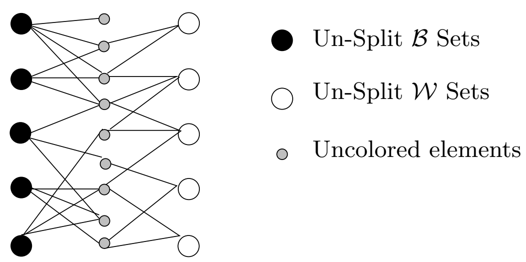

Fig. 1. Auxiliary Graph

The graph model of the subproblem may now be used to locate a crown decomposition that can be used to kernelize the subproblem instance.

By Lemma 1, if $|\mathcal{U}| \geq k$ then we can efficiently compute a nontrivial crown decomposition (C, H, J) with $C \subseteq V_{\mathcal{U}}$. Interpreting what this means for the subproblem instance, we have identified a nonempty subset $\mathcal{H}$ of the unsplit sets in $\mathcal{B} \cup \mathcal{W}$ (the head) that is matched into a subset $\mathcal{C}$ of the uncolored elements $\mathcal{U}$, the crown. Furthermore, by the properties of a crown decomposition, the elements of $\mathcal{C}$ do not belong to any other unsplit sets in $\mathcal{B} \cup \mathcal{W}$.

We can kernelize the subproblem instance according to the following rule:

Crown Reduction Rule: In the situation described above, we can reduce the subproblem instance by using the matched elements in $\mathcal{C}$ to split the sets in $\mathcal{H}$, augmenting B' and W' accordingly. Thus the reduced subproblem instance is modeled by the graph obtained by deleting the vertices that correspond to $\mathcal{C}$ and $\mathcal{H}$ and recalculating k'.

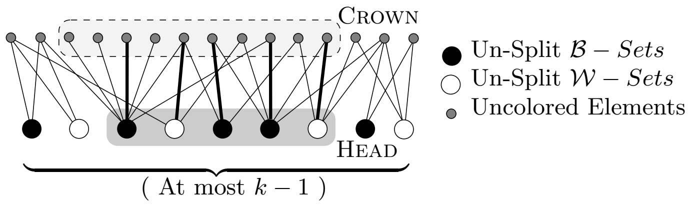

Fig. 2. Crown Decomposition

Lemma 2. *The crown rule can be used to reduce the number of elements not assigned a color to be less than* $k-1$ *in polynomial time.*

Proof. Lemma 1 states that if a graph $G = (V, E)$ has an independent set $I \subseteq V(G)$ such that $|N(I)| < |I|$ then G admits a nontrivial crown decomposition where the crown set is a subset of I. As long as the number of elements not assigned a color is greater than $|\mathcal{W} \cup \mathcal{B}|$ we can find a crown in polynomial time. Thus as $|\mathcal{W} \cup \mathcal{B}| \leq k-1$, by continually applying the crown rule we can reduce the number of elements that still need to be assigned a color to $k-1$.

3.3 Complexity Analysis of the Algorithm

Theorem 1 *The* SET SPLITTING *problem for parameter* k *can be solved in* $\mathcal{O}^*(8^{k-1})$ *time.*

Proof. Finding a maximal witness structure can be performed in $\mathcal{O}(n)$ time. By Lemma 3, $|B \cup W| \leq 2(k-1)$. The algorithm will branch into at most 4^{k-1} subproblems. Each branch is a completely contained subproblem with the partial splitting of the base set depending on which branch we are on. The Crown Rule kernelization results in a subproblem kernel having at most $k-1$ uncolored elements. Thus there are at most 2^{k-1} colorings of these to be explored. With at most 4^{k-1} subproblems, each yielding (after kernelization) at most 2^{k-1} branches to explore, we get an $\mathcal{O}^*(8^{k-1})$ FPT algorithm.

4 A New $2k$ Kernelization for Vertex Cover

If we apply the iterative compression technique to the VERTEX COVER problem, then we are concerned with the following *solution compression* form of the problem, which is specified constructively, rather than as a decision problem:

Input: A graph $G = (V, E)$ and a $(k + 1)$-element vertex cover $V' \subseteq V$.
Parameter: k
Output: Either a k-element vertex cover, or NO if none exists.

Lemma 1 guarantees a nontrivial crown decomposition if the number of vertices in $V - V'$ exceeds $k + 1$. Thus we immediately obtain a problem kernel having at most $2(k + 1)$ vertices. This improves the $3k$ kernel based on crown reduction described in [CFJ04].

Note. The astute reader will note that this is not a "kernel" in the usual sense of the word (which is generally taken in the sense of many:1 polynomial-time reductions). Here the $2k$ kernel that we achieve is actually a series of n kernels, which can be formalized as a Turing form of parameterized problem kernelization.

5 Conclusions and Open Problems

We have described some new approaches in the design of FPT algorithms, and have applied these to two concrete problems. The more substantial of these applications is an $\mathcal{O}^*(8^k)$ FPT algorithm for SET SPLITTING, significantly improving the best previous $\mathcal{O}^*(72^k)$ algorithm. While our contribution in the case of VERTEX COVER is really little more than a small observation, it is still somewhat surprising that after so much attention to this problem there is *anything* new to be said about it. Whether $2k$ kernelization via iterative compression and crown reduction has any practical advantages over Nemhauser-Trotter kernelization is an interesting question for further research along the lines of [ACFLSS04], where it is demonstrated that crown reduction is indeed useful in a practical sense. In general, it seems that the articulation of the basic toolkit for FPT algorithm design is still, surprisingly, in its infancy.

Acknowledgement. We thank Daniel Marx for helpful discussions and suggestions, particularly about the iterative compression technique, for which he suggested the name.

References

[ACFLSS04] F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the Vertex Cover problem: theory and experiments, in *Proceedings ALENEX'04,* ACM/SIAM, 2004.

[AS00] A. A. Ageev and M.I. Sviridenko. An approximation algorithm for hypergraph max k-cut with given sizes of parts, in *Proceedings of the European Symposium on Algorithms (ESA) 2000*, Springer-Verlag, *Lecture Notes in Computer Science* 1879 (2000), 32–41.

[AYZ95] N. Alon, R. Yuster, and U. Zwick. Color-Coding, in *Journal of the ACM*, 42 (1995), 844-856.

[AE97] G. Andersson and L. Engebretsen. Better approximation algorithms for set splitting and Not-All-Equal SAT, in *Information Processing Letters*, Vol. 65, pp. 305-311, 1998.

[CFJ04] B. Chor, M. Fellows, and D. Juedes. Linear Kernels in Linear Time, or How to Save k Colors in $O(n^2)$ Steps). *Proceedings WG 2004 - 30th Workshop on Graph Theoretic Concepts in Computer Science*, Springer-Verlag, *Lecture Notes in Computer Science*, 2004 (to appear).

[CFJK01] J. Chen, D. K. Friesen, W. Jia and I. A. Kanj. Using Nondeterminism to Design Efficient Deterministic Algorithms, in *Proceedings 21st Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Springer-Verlag, *Lecture Notes in Computer Science* 2245 (2001), 120–131. (Journal version to appear in *Algorithmica*.)

[DFR03] F. Dehne, M. Fellows, and F. Rosamond. An *FPT* Algorithm for Set Splitting, in *Proceedings WG 2003 - 29th Workshop on Graph Theoretic Concepts in Computer Science*, Springer-Verlag, *Lecture Notes in Computer Science* 2880 (2003), 180-191.

[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[F03] M. Fellows. Blow-ups, Win/Win's and Crown Rules: Some New Directions in *FPT*, in *Proceedings WG 2003 - 29th Workshop on Graph Theoretic Concepts in Computer Science*, Springer-Verlag, *Lecture Notes in Computer Science* 2880 (2003), 1-12.

[FKN04] M. Fellows, C. Knauer, N. Nishimura, P. Ragde, F. Rosamond, U. Stege, D. Thilikos and S. Whitesides. Faster Fixed-Parameter Tractable Algorithms for Matching and Packing Problems. *Proceedings of the European Symposium on Algorithms (ESA) 2004*, Springer-Verlag, *Lecture Notes in Computer Science*, 2004 (to appear).

[FHRST04] M. Fellows, P. Heggernes, F. Rosamond, C. Sloper, and J.A. Telle. Exact Algorithms for Finding k Disjoint Triangles in an Arbitrary Graph, to appear in *Proceedings WG 2004 - 30th Workshop on Graph Theoretic Concepts in Computer Science*, Springer-Verlag, *Lecture Notes in Computer Science*, 2004 (to appear).

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[JZC03] W. Jia, C. Zhang, and J. Chen. An Efficient Parameterized Algorithm for Set Packing. Manuscript, 2003, to appear in *Journal of Algorithms.*

[Ma04] D. Marx. Chordal Deletion is Fixed-Parameter Tractable. Manuscript, 2004.

[MPS04] L. Mathieson, E. Prieto, and P. Shaw. Packing Edge Disjoint Triangles: A Parameterized View. *Proceedings of the International Workshop on Parameterized and Exact Computation*, Springer-Verlag, *Lecture Notes in Computer Science* (this volume), 2004.

[Nie02] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms,* Habilitationschrift, University of Tubingen, 2002.

[Pe94] E. Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4 (1994), 133–157.

[PS03] E. Prieto and C. Sloper. Either/Or: Using Vertex Cover Structure in Designing *FPT* Algorithms–the Case of k-Internal Spanning Tree. *Proceeding of the Workshop on Algorithms and Data Structures (WADS) 2003*, Springer-Verlag, *Lecture Notes in Computer Science* 2748 (2003), 474-483.

[PS04] E. Prieto and C. Sloper. Looking at the Stars. *Proceedings of the International Workshop on Parameterized and Exact Computation*, Springer-Verlag, *Lecture Notes in Computer Science* (this volume), 2004.

[RSV03] B. Reed, K. Smith, and A. Vetta. Finding Odd Cycle Transversals. *Operations Research Letters* 32 (2004), 299–301.

[Woe03] G. J. Woeginger. Exact Algorithms for NP-Hard Problems: A Survey. *Proceedings of 5th International Workshop on Combinatorial Optimization-Eureka, You Shrink! Papers dedicated to Jack Edmonds,* M. Junger, G. Reinelt, and G. Rinaldi (Festschrift Eds.) Springer-Verlag, *Lecture Notes in Computer Science* 2570 (2003), 184-207.

[ZL01] H. Zhang and C.X. Ling. An Improved Learning Algorithm for Augmented Naive Bayes. *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Springer-Verlag, *Lecture Notes in Computer Science* 2035 (2001), 581–586.