Resampling in an Indefinite Database to Approximate Functional Dependencies

Ethan Collopy and Mark Levene
Email:{ecollopy, mlevene}@cs.ucl.ac.uk

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, U.K.

Abstract. Functional Dependency satisfaction, where the value of one attribute uniquely determines another, may be approximated by Numerical Dependencies (NDs), wherein an attribute set determines at most k attribute sets. Hence, we use NDs to "mine" a relation to see how well a given FD set is approximated. We motivate NDs by examining their use with indefinite information in relations. The family of all possible ND sets which may approximate an FD set forms a complete lattice. Using this, a proximity metric is presented and used to assess the distance of each resulting ND set to a given FD set.
Searching for a definite relation extracted from an indefinite relation which satisfies a given set of FDs, known as the consistency problem, has been shown to be NP-complete. We propose a novel application of the bootstrap, a computer intensive resampling technique, to determine a suitable number of definite relations upon which to apply a heuristic based hill-climbing algorithm which attempts to minimise the distance between the best ND set and the given FD set. The novelty is that we repeatedly apply the bootstrap to an indefinite relation with an increasing sample size until an approximate fixpoint is reached at which point we assume that the sample size is then representative of the indefinite relation. We compare the bootstrap with its predecessor, the jackknife, and conclude that both are applicable with the bootstrap providing additional flexibility. This work highlights the utility of computer intensive resampling within a dependency data mining context.
Key Words - Functional Dependency, Numerical Dependency, Data Mining, Indefinite Relation, Resampling, Bootstrap

1 Introduction

Numerical Dependencies (NDs)[1] are generalisations of Functional Dependencies (FDs) which allow an attribute set to uniquely determine up to k different attribute set values, noting that $k = 1$ in the case of FDs. Indefinite information representation in relations has been shown to be a useful facility for incomplete specifications in design and planning applications [2]. We define *indefinite cells* as cells containing one or more values which represent a set of possibilities denoting the current limit of knowledge in the database. A definite relation extracted from one containing indefinite information is a relation with the same schema and definite cells, which are invariant throughout, but with each indefinite cell,

say C, replaced with a definite cell containing one value from C. Associated with an indefinite relation may be a set of integrity constraints. There may be cases, highlighted below, where the traditional FD is too strict and a *weaker* integrity constraint, such as an ND, is required. Table 1 shows how we might want to represent indefinite information in a teaching relation $PLAN$(Lecturer,Course). Irrespective of whatever courses Mark and Robin decide to teach no definite relation extracted from $PLAN$ will satisfy the FD $Lecturer \rightarrow Course$ though all satisfy the ND $Lecturer \rightarrow^2 Course$, representing that a Lecturer can teach up to two courses in a year. For an FD $X \rightarrow Y$, the set of all possible NDs, $X \rightarrow^k Y$, which may approximate this allow k to range from 1 up to the maximum active domain size (ADS) combination in Y. All of these possibilities are shown to form a complete lattice which is then used as the base for a metric on ND sets which we use to gain a value between 0 and 1 for the proximity of a relation to FD set satisfaction.

Table 1. An indefinite relation $PLAN$

Lecturer	Course
Mark	{B11a,C320}
Robin	B11a
{Robin,Mark}	B151

Given a set of FDs F and an indefinite relation r (a relation with one or more indefinite cells) we tackle the problem of attempting to find a definite relation extracted from r which satisfies F. This is widely known as the *consistency problem*, shown to be NP-Complete in general, and of polynomial time complexity in the case where indefinite information is only allowed in attributes which are present in the right hand side of FDs (referred to as a *good* database) or when the FDs have a singleton right hand side and attributes of at most arity two are allowed in the left hand side [2]. Within our algorithm we use the chase process [3], a heuristic designed to modify a database to satisfy constraints, extended in [4] for numerical dependencies and indefinite information, and used in a hill-climbing fashion. Henceforth, we refer to definite relations as *possible worlds*.

We use the bootstrap procedure [5], a computationally intensive statistical resampling procedure that requires no assumptions on the distribution of the possible worlds. We initially take a sample S of n observed possible worlds. Based upon this sample we perform a number of bootstrap replications. Each bootstrap replication, of size n, samples from S with replacement. In this way the bootstrap can be used to provide a guide to the distribution of ND satisfaction in the possible worlds. The key assumption we make in this case is that our sample of observed possible worlds is representative of the indefinite relation. We repeat the bootstrap with an increasing sample size of observed possible worlds. After each bootstrap iteration we calculate the mean and standard error. The number of observed possible worlds (sample size) is increased until the bootstrap procedure converges to an approximate fixpoint, defined as the state where the change in variance is sufficiently small. In this sense the convergence of the bootstrap mean value tells us, with a high probability, that increasing the sample size further

will not provide us with any additional information concerning the distribution of data within the indefinite relation. Our results have shown this convergence always occurs at a sample size that is an upper bound on the number actually used by the chase hill-climbing procedure. This is a novel application of sampling within databases, not previously used within the limits of our experience.

We also experimented with the jackknife resampling technique for comparison purposes. The jackknife creates n resamples from an original sample of size n where each resample is of size $n-1$ with a single possible world left out of each resample. Given the restricted choice of points within the jackknife resamples the returned variance is smaller, on average, than that obtained from the bootstrap and it reaches a fixpoint with a fewer number of worlds. As anticipated, the difference between the bootstrap and the jackknife is minimal. The jackknife was shown to approximate the bootstrap in [5] though we conclude that the bootstrap is generally superior in its role of parameter estimation, providing a better but not excessive parameter for a suitable sampling size as well as being more flexible. We conducted simulations to test the viability of our approach. These are described extensively in [4] and indicate that our use of the bootstrap for parameter setting is a useful tool.

The rest of the paper is organised as follows. In Section 2 we introduce the concepts of indefinite information and numerical dependency, which is central to the process of approximating the distance to FD sets, as well as the background on the lattice of NDs and the proximity metric. Section 3 introduces the framework for the bootstrap and in Section 4 we describe and analyse the bootstrap and jackknife algorithms. Finally, in Section 5 we give our concluding remarks.

2 Relational Database Background

Definition 1 (Relation schema and indefinite relations). Let $\mathcal{U}$ be a countable set of attributes and $\mathcal{D}$ be a countable set of domain values. A *relation schema* R is a finite set of attributes in $\mathcal{U}$. An *(indefinite) tuple* t over R is a total mapping from R into $\mathcal{P}(\mathcal{D})$ such that $\forall A \in \mathrm{R}$, $t(A) \in \mathcal{P}(\mathcal{D})$. A tuple t over R is *definite* if $\forall A \in \mathrm{R}$, $\mid t(A) \mid = 1$, i.e. $t(A)$ is a singleton, where $\mid t(A) \mid$ denotes the cardinality of $t(A)$.

A *indefinite relation* r over R is a finite (possibly empty) set of indefinite tuples over R. A relation over R is *definite* if all of its tuples are definite. The set of all possible worlds which may be formed from r is precisely the set of all combinations of replacing each indefinite cell with one of its values. From now on we let R *be a relation schema*, r *be a relation over* R *and* $t \in r$ *be an indefinite tuple*. Letters from the beginning of the alphabet such as A, B denote singleton attribute sets {A},{B} in R. We generalise the concept of an FD by a *numerical dependency*.

Definition 2 (Numerical dependency). A *numerical dependency* over R (or simply an ND) is a statement of the form $\mathrm{X} \rightarrow^k \mathrm{Y}$, where X, Y $\subseteq$ R and $k \geq 1$. $\mathrm{X} \rightarrow^k \mathrm{Y}$ is satisfied when for each unique attribute set value in X there are at most k different attribute set values in Y. A set of NDs N is *satisfied* in s, denoted by $s \models \mathrm{N}$, whenever $\forall\ \mathrm{X} \rightarrow^k \mathrm{Y} \in \mathrm{N}$, $s \models \mathrm{X} \rightarrow^k \mathrm{Y}$.

From now on we let N *be a set of NDs over* R, F *a set of FDs over* R, and $X \rightarrow^k Y$ *be a single ND over* R, with $k \geq 1$. When $k = 1$, $X \rightarrow^1 Y$ is an FD, written as $X \rightarrow Y$. A set of FDs F is *weakly* satisfied (or simply satisfied whenever no ambiguity arises) in a relation r, denoted by $r \models\!\!\approx F$, whenever r has a possible world s such that $s \models F$. If $r \models\!\!\approx F$ we say that r is *consistent* with respect to F; otherwise if $r \not\models\!\!\approx F$ then we say that r is *inconsistent* with respect to F (or simply r is inconsistent). We define a set of NDs N to be weakly satisfied in a relation r in the same way as for FDs; similarly we define a relation r to be consistent with respect to a set of NDs if $r \models\!\!\approx N$ and otherwise to be inconsistent. We note that if $r \models\!\!\approx X \rightarrow^k Y$ then it is also the case that $r \models\!\!\approx X \rightarrow^{k+1} Y$, i.e. the smaller k the *more functional* the ND. We consider, without loss of generality, only FDs and NDs with singleton right hand sides.

Definition 3 (The consistency problem). Given a set of FDs F and a relation r, possibly containing indefinite cells, the *consistency problem* is the problem of deciding whether $r \models\!\!\approx F$.

Definition 4 (More functional set of NDs). A set of NDs N_1 over R is *more functional* than a set of NDs N_2 over R, denoted by $N_2 \sqsubseteq N_1$, whenever $X \rightarrow^{k_2} A \in N_2$ if and only if $X \rightarrow^{k_1} A \in N_1$ and $k_1 \leq k_2$.

The set-theoretic relation, more functional than, is a partial order in the sets of NDs. Assume that we are considering only sets of NDs which are more functional than a given set of NDs, N over R, each of the form $X \rightarrow^k Y$, for some $k \geq 1$. Then the family of sets of NDs that are more functional than N form a lattice whose bottom element is N and whose top element is the set of FDs induced by N, i.e. $\{X \rightarrow Y \mid X \rightarrow^k Y \in N\}$. The *least upper bound*, *lub*, of N_1 and N_2 is the set of NDs $\{X \rightarrow^{min(k_1,k_2)} Y \mid X \rightarrow^{k_1} Y \in N_1 \text{ and } X \rightarrow^{k_2} Y \in N_2\}$, where $min(k_1, k_2)$ is the minimum of k_1 and k_2, and the *greatest lower bound*, *glb*, is defined similarly using maximum. We call the lattice, whose top element is the set of FDs F over R and whose bottom element is the set of NDs $\{X \rightarrow^m Y \mid X \rightarrow Y \in F\}$, $\mathcal{L}_m(F)$ (or simply $\mathcal{L}_m$ if F is understood from context), with $m \geq 1$. Therefore, we can *approximate* a set of FDs F by a set of NDs N such that $N \sqsubseteq F$. The *closer* N is to F in $\mathcal{L}_m$ the better the approximation is. From now on we let $\mathcal{L}_m$ *be the lattice of NDs whose top element is* F and assume that $| r | = m + 1$, with $m \geq 1$. A set of NDs N over R is the *best approximation* of a set of FDs F over R with respect to a relation r over R, with $| r | = m + 1$ (or simply the best approximation of F if r is understood from context), if $r \models\!\!\approx N$ and there does not exist a set of NDs, $N' \in \mathcal{L}_m$ such that $N \prec N'$ and $r \models\!\!\approx N'$.

We introduce a measure for calculating the proximity of two ND sets using their position within the lattice. We show in [4] that this measure isa metric. We define the *size* of a set of NDs N to be the number of attributes appearing in N including repetitions and define a *step*, either up or down, to be exactly minus or plus one, respectively, to a single branch of one ND within an ND set. Furthermore, we say that N_2 is *covered by* N_1, denoted by $N_2 \prec N_1$, where $N_1, N_2 \in \mathcal{L}_m$, if $N_1 \neq N_2$, $N_2 \sqsubseteq N_1$ and $\forall N' \in \mathcal{L}_m$ such that $N_2 \sqsubseteq N' \sqsubseteq N_1$ we have $N' = N_2$. In our simulations one of the ND sets is always the given FD set F in which case the metric tells us the proximity between the ND set and F. We

define *distance* to be the sum of *steps* taken in the lattice. We define the bottom of the lattice to be the set of NDs with each branching factor equivalent to the domain size of the attribute on the right hand side of each ND.

Proposition 5. The maximum distance between any two points in the lattice to their *lub* is always equivalent to the distance from the bottom to the top of the lattice.
Proof. By induction, presented in [4]. □

Definition 6 (Proximity of two ND sets). Given two sets of NDs N_1 and N_2 we define the metric as follows:

$$p(N_1, N_2) = \frac{\Sigma_{i=1,2} \text{ Distance from } N_i \text{ to } lub\{N_1, N_2\}}{\text{Max distance between any two ND sets to their } lub \text{ in the lattice}}$$

Fig. 1. Average Number of Worlds given as upper bounds by the Bootstrap and Jackknife techniques

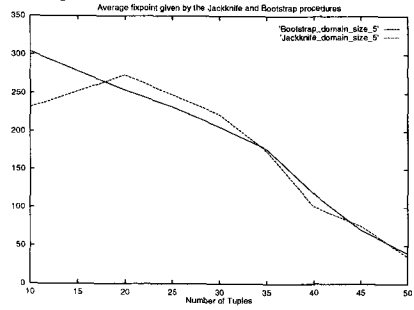
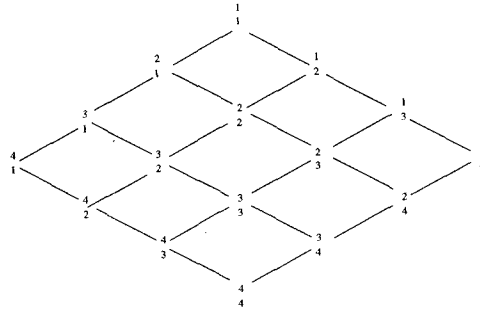


Fig. 2. Lattice of NDs for a relation of 2 FDs (not specified) and an ADS of 4 for each dependency



3 Incorporating the Bootstrap and the Jackknife

The bootstrap [5] is a data driven simulation method for non-parametric statistical inference. Given that the number of possible worlds of an indefinite relation increases exponentially in the size of the relation it is impossible to examine all possible worlds for the best solution. The complete population distribution is unknown; otherwise we would know exactly how many definite relations to generate to have a specific probability of finding the closest ND set to the given FD set. This suggests applying a bootstrap procedure to a sample of definite instances to approximate the population distribution based on the sample distribution [5]. We use the bootstrap procedure to tell us how many worlds we need to consider so that we have a high confidence that generating additional worlds will not improve our solution. Algorithm 1 presents this procedure and Algorithm 2 presents a corresponding procedure using the jackknife.

Definition 7 (The Bootstrap Sample). Given an indefinite relation r over schema R we uniformly randomly extract n possible worlds. Each of these worlds will satisfy a set of NDs (which may contain FDs). These n possible worlds are referred to as the original sample or *observed possible worlds* and are written as $\tilde{p} = (r_1, r_2, \ldots, r_n)$. A bootstrap sample is $\tilde{p}^\star = (r_1^\star, r_2^\star, \ldots, r_n^\star)$ where for all $i = 1, 2 \ldots, n$ each $r_i^\star$ is randomly selected with replacement from the n observed possible worlds in $\tilde{p}$.

We denote the q NDs which may hold in r by $X_i \rightarrow^{k_i} Y_i$ where $1 \leq i \leq q$ and refer to the branching factor k which holds for ND $X_i \rightarrow^{k} Y_i$ in r as $br_{X_i Y_i}(r)$. When we refer to the sample mean of a set of possible worlds we are implying the sample mean of the sets of NDs of the possible worlds.

Definition 8 (The Bootstrap Sample Mean). Given a bootstrap sample $\tilde{p}^\star = (r_1^\star, r_2^\star, \ldots, r_n^\star)$, we calculate the mean $\bar{s}(\cdot)$, or any other statistic of interest, in exactly the same way as we would have for the original sample of ND sets, each containing m NDs, $\bar{s}(\tilde{p}^\star) = \{K_j \mid 1 \leq j \leq m\}$ where $K_j = \sum_{i=1}^{n} \frac{br_{X_j Y_j}(r_i^\star)}{n}$

Definition 9 (The Bootstrap Mean of all Values). Given a set of B bootstrap samples $\tilde{p}_b^\star$, we calculate the mean $\bar{s}(\cdot)$, or any other statistic of interest, in exactly the same way as we would have for the original sample, $\bar{s}(\tilde{p}_b^\star) = \Sigma_{i=1}^{B} \bar{s}(\tilde{p}_i^\star)/B$.

Algorithm 1 (BOOTSTRAP(*nd_bg*, *B*)).

```
1.begin
2.ND_m := ∅ ;
3.n := | nd_bg |;
4.for 1 to B do
5.  ND_s := Uniform Randomly select n
      ND sets from nd_bg with replacement;
6.  Insert the mean of ND_s into ND_m;
7.end for
8.return the mean of ND_m;
9.end.
```

Algorithm 2 (JACKKNIFE(*nd_bg*)).

```
1.begin
2.ND_m := ∅;
3.n := | nd_bg |;
4.for j := 1 to n do
5.  ND_s := nd_bg - nd_j;
6.  Insert the mean of ND_s into
      ND_m;
7.end for
8.return the mean of ND_m;
9.end.
```

The Bootstrap Replication Size (BRS), B in Algorithm 1, is the number of times a bootstrap sample of size n is created from the observed possible worlds and evaluated on a parameter of interest. We denote the B bootstrap samples by $\tilde{p}_b^\star = (\tilde{p}_1^\star, \tilde{p}_2^\star, \ldots, \tilde{p}_B^\star)$. [5] tackles how large the BRS should be. Given a BRS B, [5] refers to the *ideal bootstrap estimate* which takes B equal to infinity. This is not true for indefinite relations where the ideal limit is the number of possible worlds in the relation. [5] show the amount of computation time it takes for increased BRS sizes increases linearly. We show that this is also the case for increasing the BRS for indefinite relations, exemplified in [4].

Definition 10 (The Bootstrap Standard Error for Indefinite Relations). The sample standard error in the values for B bootstrapped values is:

$$\hat{se}_B = \{\frac{1}{B}\Sigma_{i=1}^{B}(\bar{s}(\tilde{p}^\star) - \bar{s}(\tilde{p}_b^\star)\}^{1/2}$$

We now describe the methods of our Bootstrap application, detailed in Algorithm 3. We start with a small initial sample size and a Bootstrap Replication Size B. Having created B bootstrap samples we will have a bootstrap mean of all values in the form of an ND set. From this value we can use the bootstrap to calculate its standard deviation. From this we can empirically infer the width of the interval in which a certain percentage of the relations occur, either using standard confidence intervals or by creating the confidence intervals empirically

using an ordering of the bootstrap resamples. We increase the sample size on each iteration by a fixed amount, δ, until we reach a point where the mean value of the NDs in the ND set stabilises. The convergence to stability is controlled by the accuracy to t significant digits, with $t = 3$ providing a sufficient accuracy in our simulations. This convergence provides a parameter whereupon anything higher is unlikely to have much additional change in variance and this is also verified by the convergence of the empirical confidence intervals. It is unlikely, even for an ND set with just one dependency, for the fixpoint to be reached randomly, and running our simulations in batches of 500 implied that any outlying fixpoint values would have a negligible impact on the final results obtained.

Algorithm 3 (WORLD_LIMIT (r, **F**, **B**)).

```
1.  begin
2.    n := initial(r); % sample size
3.    N̂_0 := Highest ND set satisfiable in r using chase ;
4.    N̂_1 := ∅; j := 1;
5.    while N̂_j, N̂_{j-1} are not approximate fixpoint do
6.      ND_bag := n ND sets from n possible worlds;
7.      N̂_j := BOOTSTRAP(ND_bag, n, B);
8.      n := n + δ; j := j + 1; % Increase the sample size by δ
9.    end while
10.   return n;
11. end.
```

We also examined the variance of the observed possible worlds, for a range of original sample sizes, as the bootstrap replication size was scaled from 20 up to 50,000 to decide on a suitable BRS. As this was increased we noted that above 1000 there was negligible change in the variance. For the purposes of our experiment setting B at 100 gave suitable results above which there was negligible change.

4 Applying Resampling to the Consistency Problem

Algorithm 3, WORLD_LIMIT(r,F,B), describes our novel use of the bootstrap procedure. Details of the simulations we conducted for different FD sets and indefinite relations are discussed in [4]. Our procedure relies on the assumption that different sample sizes are required proportional to the variance within an indefinite relation in the different ND sets which may be satisfied in possible worlds. The number of dependencies in the given FD set also influences the results obtained from our use of the bootstrap. We use the BOOTSTRAP algorithm in exactly the same manner as a standard bootstrap procedure despite that we potentially have all possible worlds within the indefinite relation. Based on this we conducted experiments whereby the bootstrap resamples were obtained not from the original sample but from the indefinite relation. The variance of resampling from the relation was much higher than resampling from the sample and in such cases the upper bound was much higher. Therefore, based on our results, we conjecture that it is suitable to use just one original sample from the indefinite relation within each iteration of WORLD_LIMIT.

In Figure 4 we see that, for both FD set $F_1 = \{A \rightarrow B, A \rightarrow C, A \rightarrow D\}$ and $F_2 = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$, as the number of tuples increases there is a

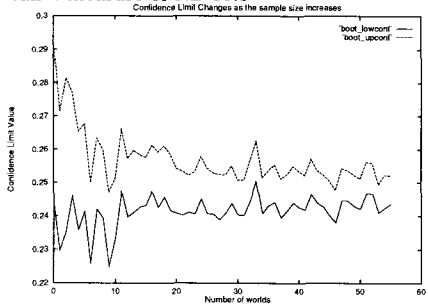


Fig. 3. Efron's empirical percentile confidence limits shown to converge for the distance measure of ND sets

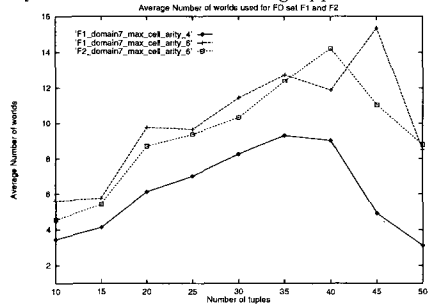


Fig. 4. Average Number of Worlds required by the chase and hill-climbing approach

slight peak, after which further increases in the number of tuples results in a fall in the average number of worlds required. This is due to every relation within a batch having a fixed domain size d and a maximum indefinite cell arity, reaching a point where it is likely that any further increases in the tuple size will lead to the satisfaction of the numerical dependency set with each branch determining up to d branches and so fewer worlds are required before any attempts to apply the chase returns an undefined relation implying that nothing better can be found. The peaks in Figure 4 were reflected in the values of α returned by our bootstrap technique. In our application of the bootstrap, as the relation size of a random relation is increased and the domain size is held constant, the sampling will also reach a point where the variance in the samples amongst the randomly generated possible worlds is reduced due to most possible worlds satisfying the NDs each with a branching factor close to their domain size.

The question of why the bootstrap provides an upper bound remains. The chase and hill-climbing algorithm exits if the chase heuristic returns an undefined relation for the current highest found ND set $N_\top$ in the lattice. This implies that the indefinite relation is unable to satisfy any ND sets above $N_\top$. Given that this generally occurs before reaching the limit α (provided by the bootstrap) it seems reasonable to propose that the variance across the possible worlds of an indefinite relation, in terms of ND set satisfaction, is a naive statistic and our hill-climbing and chase heuristic method is sufficient to reach a *good* approximation before examing α initial points. The correspondence between the heuristic and the changing upper limit, due to changing variance of ND set satisfaction in indefinite relations, is to be expected and its usefulness is highlighted in this work.

4.1 Differences between resampling methods

The strategy of the jackknife is to remove a single data point from each resample. This allows the creation of n jackknife resamples from an original sample of size n. The bootstrap provides additional flexibility in that the sample is made up of any values uniformly and randomly selected with replacement from the original and, additionally, is not limited to n resamples. In our process the number of worlds required is increased until a fixpoint is reached. Using the jackknife as the worlds reach a large number q we are constrained to q resamples, each of size $q-1$. Under the bootstrap application we have a fixed number of resamples

which, in the majority of cases, will increase to a sample size that is smaller than the q required by the jackknife. We found that the results were very similar for both the bootstrap and jackknife, highlighted in Figure 1, despite our use of the bootstrap conducting fewer replications than the jackknife at large sample sizes. Figure 1 also presents the falling limit of the fixpoint as the domain size is held constant but the tuple size increases, due to a reduction in variance within possible worlds as the relation size grows, highlighted in Figure 3 where the empirical confidence limits for the bootstrap process are shown to converge for the distance measure of an ND set.

5 Conclusion

We have described how the representation of indefinite information lends itself to utilising ND sets. In addition to this we note that NDs suitably approximate FDs in a data mining context. In many dependency data mining applications, which range from data summarisation to learning within decision trees [6], we may wish to obtain a numerical value, between 0 and 1, denoting how close a set of FDs are to being satisfied; the metric presented in this paper achieves this. In [2] we are shown how indefinite information may be used to represent a possible schedule. Our approach allows us to discover an approximation to an *ideal* relation, that which satisfies a set of FDs. NDs are a useful tool in this context and schedule representation within relational databases is enhanced with their use. The consistency problem for relations with indefinite information is widely known to be *NP-complete*. Therefore we cannot expect to develop a polynomial time based solution unless $P = NP$ or the database is restricted as in [2]. Our approach does however introduce an interesting new technique based on sampling, incorporating the bootstrap to provide useful approximations for problems such as the consistency problem. Simulations imply that the bootstrap provides a suitable upper bound. We are also planning to explore resampling within the temporal database domain, another area where there is a combinatorial explosion of data points.

References

1. J. Grant and J. Minker. Inferences for numerical dependencies. *Theoretical Computer Science*, 41:271–287, 1985.
2. K. Vadaparty and S. Naqvi. Using constraints for efficient query processing in nondeterministic databases. *IEEE Transactions on Knowledge and Data Engineering*, 7(6):850 – 864, 1995.
3. H. Mannila and K-J. Räihä. *The Design of Relational Databases*. Addison–Wesley, 1992.
4. E. Collopy and M. Levene. Using numerical dependencies and the bootstrap for the consistency problem. Technical Report RN/98/2, University College London, U.K., 1998.
5. B. Efron and R. Tibshirani. Bootstrap methods for standerd errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):54 – 77, 1986.
6. G. Piatetsky-Shapiro and C. J. Matheus. Measuring data dependencies in large databases. In *Proceedings of the Workshop on Knowledge Discovery in Databases*, pages 162–173, Washington DC, 1993.