

Aggregation of Imprecise and Uncertain Information for Knowledge Discovery in Databases

Sally McClean, Bryan Scotney and Mary Shapcott

School of Information and Software Engineering, University of Ulster
Coleraine, BT52 1SA, Northern Ireland
{si.mcclean, bw.scotney, cm.shapcott}@ulst.ac.uk

Abstract

We consider the problem of *aggregation* for *uncertain* and *imprecise* data. For such data, we define aggregation operators and use them to provide information on properties and patterns of data attributes. The aggregates that we define use the Kullback-Leibler *information divergence* between the aggregated probability distribution and the individual tuple data values. We are thus able to provide a probability distribution for the domain values of an attribute or group of attributes using imperfect data.

Information stored in a database is often subject to uncertainty and imprecision. An extended relational data model has previously been proposed for such data which allows us to quantify our uncertainty and imprecision about attribute values by representing them as a probability distribution. Our aggregation operators are defined on such a data model. The provision of such operators is a central requirement in furnishing a database with the capability to perform the operations necessary for Knowledge Discovery in Databases.

Background

Frequently, real life data are *uncertain*, i.e. we are not certain about the truth of an attribute value, or *imprecise*, i.e. we are not certain about the specific value of an attribute. Such imprecision might occur naturally as a result of data being provided at different levels of the concept hierarchy or from the integration of distributed databases. It is therefore important that appropriate functionality is provided for generalised database systems to provide intelligent ways of handling such imperfect information. A recent survey of various approaches to handling imperfect information in Data and Knowledge Bases has been provided by Parsons (1996).

A database model which is based on *partial values* and *partial probabilities* (DeMichiel, 1989; Tseng et al., 1993;

Chen and Tseng, 1996) has been proposed to handle both imprecise and uncertain data. Partial values may be thought of as a generalisation of null values: rather than not knowing anything about a particular attribute value, we may identify the attribute value as belonging to a set of possible values. *Partial probabilistic values* (Chang and Chen, 1994) further generalise by assigning probabilities to the partial values. We may thus combine the imprecision-handling capacity of partial values with the uncertainty-handling capabilities of probability theory. This capacity to handle both imprecise and uncertain data is a major strength of our current approach. The facility to aggregate such data is a central requirement in providing a database with the capability to perform the operations necessary for Knowledge Discovery in Databases, where we are frequently concerned with identifying interesting attributes or interesting relationships between attributes.

Data Models for Imprecise and Uncertain Data

In a generalised Relational Database we consider an attribute A and tuple t of a relation R which has values which are imprecise in that in any particular tuple an attribute value may be a partial value.

Definition 1. A *partial value* is determined by a set of possible attribute values of tuple t of attribute A of which one and only one is the true value. We denote a partial value η by $\eta = [a_r, \dots, a_s]$ corresponding to a set of h possible values $\{a_r, \dots, a_s\}$ of the same domain, in which exactly one of these values is the true value of η . Here, h is the cardinality of η ; $\{a_r, \dots, a_s\}$ is a subset of the domain set $\{a_1, \dots, a_k\}$ of attribute A of relation R and $h \leq k$.

We extend the definition to a *partial probability distribution*, where probabilities are assigned to each partial value.

Definition 2. A *partial probability distribution* is a vector of probabilities $\phi(\eta) = (p_1, \dots, p_e)$ which are associated with a partition of partial values $\eta = (\eta_1, \dots, \eta_e)$ of tuple t of attribute A such that $p_i =$ probability (the attribute value is

a member of partial value set η_i). Here $\sum_{i=1}^e p_i = 1$.

Example 1: For a partial probability distribution to be defined on the attribute SMOKING_STATUS, we have a set of partial values which forms a partition of the domain, such as ({‘heavy’, ‘light’}, {‘ex-heavy’, ‘ex-light’}, {‘never’}). We then define a partial probability distribution on this partition, e.g. <{‘heavy’, ‘light’}, 0.4>, <{‘ex-heavy’, ‘ex-light’}, 0.35>, <{‘never’}, 0.25>.

This distribution means, for example, that the probability of being either a heavy or a light smoker is 0.4.

An extended relational model in which the tuples consist of partial values was described in DeMichiel (1989) and Chen and Tseng (1996). We extend this data model to an extended relational database model based on a partial probability distribution. An equivalent data model, along with some extended relational operators such as *project* and *join*, has been proposed by Barbará et al. (1992) who introduced the term *probability data model* (PDM).

ID	SEX	SMOKING STATUS	HYPER TENSION	HEART DISEASE
01	<{M},0.0> <{F},1.0>	<{S},1.0> <{E},0.0> <{N},0.0>	<{S,M},1.0> <{N},0.0>	<{S},0.75> <{M,N},0.25>
02	<{M},1.0> <{F},0.0>	<{S},0.0> <{E},1.0> <{N},0.0>	<{S,M,N},1.0>	<{S},0.75> <{M},0.2> <{N},0.05>
03	<{M},0.0> <{F},1.0>	<{S},0.0> <{E},0.2> <{N},0.8>	<{S,M},0.0> <{N},1.0>	<{S,M},0.0> <{N},1.0>
04	<{M},1.0> <{F},0.0>	<{S},0.9> <{E},0.1> <{N},0.0>	<{S},1.0> <{M,N},0.0>	<{S},1.0> <{M,N},0.0>
05	<{M},0.0> <{F},1.0>	<{S},0.0> <{E},1.0> <{N},0.0>	<{S,M},0.0> <{N},1.0>	<{S},0.2> <{M,N},0.8>
06	<{M},1.0> <{F},0.0>	<{S},1.0> <{E},0.0> <{N},0.0>	<{S,M},1.0> <{N},0.0>	<{S},1.0> <{M,N},0.0>
07	<{M},1.0> <{F},0.0>	<{S},0.0> <{E},0.0> <{N},1.0>	<{S,M},0.0> <{N},1.0>	<{S},0.1> <{M},0.2> <{N},0.7>
08	<{M},1.0> <{F},0.0>	<{S},1.0> <{E},0.0> <{N},0.0>	<{S},1.0> <{M,N},0.0>	<{S},0.9> <{M},0.05> <{N},0.05>
09	<{M},0.0> <{F},1.0>	<{S},0.75> <{E},0.25> <{N},0.0>	<{S,M},1.0> <{N},0.0>	<{S},0.0> <{M},1.0> <{N},0.0>
10	<{M},1.0> <{F},0.0>	<{S},1.0> <{E},0.0> <{N},0.0>	<{S,M},0.0> <{N},1.0>	<{S},0.2> <{M,N},0.8>

Table 1. A Probability Data Table

Table Legend

SEX: M - male, F - female;

SMOKING_STATUS: S - smoker, E - ex - smoker, N - never;

HYPERTENSION: S - severe, M - mild, N - no hypertension;

HEART_DISEASE: S - severe, M - mild, N - none.

Definition 3. A *probability data model* (or table) \mathbf{R} is a relation based on the probability distribution of partial values for domains D_1, D_2, \dots, D_n of attributes A_1, A_2, \dots, A_n where $\mathbf{R} \subseteq P_1 \times P_2 \times \dots \times P_n$ and P_i is the set of all the probability distributions on the power set of domain D_i . Each element of the table is a probability distribution specified on a partition of the appropriate domain into partial values.

An example of a probability data model is presented in Table 1. This general relational data model allows us to represent a number of other models as special cases. For example, the attribute SEX is crisp with no null values, i.e. a value is M with probability 1 or F with probability 1. The attribute SMOKING_STATUS consists entirely of crisp probabilistic values, i.e. in each case we know probabilities for each of the three possible values - smoker, ex-smoker or never smoked. The attribute HYPERTENSION consists of true partial values: here attribute values are imprecise but not uncertain. The final attribute, HEART_DISEASE, is a true partial probability distribution; the values are therefore both imprecise and uncertain.

Aggregation of Partial Probabilities

In this section we develop an approach which allows us to aggregate attributes of a partial probability distribution relation such as that presented in Table 1. Thus this general aggregation operator (*gagg*) must include, as special cases, crisp and certain data (e.g. SEX in Table 1) and crisp and uncertain data (e.g. SMOKING_STATUS).

Notation: As before, we consider an attribute A_j of a partial probability relation \mathbf{R} with corresponding domain $D_j = \{v_1, \dots, v_k\}$ which has tuples t_1, \dots, t_m . The value of the r th tuple of A_j is a probability distribution described by vectors

$$t_r \cdot A_j = (f_{r1}^{(j)}, \dots, f_{rg_r}^{(j)}) \quad \text{and} \quad \mathbf{P} = (S_{r1}^{(j)}, \dots, S_{rg_r}^{(j)}),$$

where \mathbf{P} is a partition of the domain D_j and $f_{r\ell}^{(j)} = \text{Prob}(\text{tuple value is a member of } S_{r\ell}^{(j)})$; g_r is the number of sets in the partition for tuple t_r . We further define:

$$q_{ir\ell} = \begin{cases} 1 & \text{if } v_i \in S_{r\ell}^{(j)} \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4: The *general aggregate* of a number of probability distribution values on attribute A_j of relation \mathbf{R} , denoted $gagg(\mathbf{R}, A_j)$, is defined as a vector-valued function:

$agg(\mathbf{R}, A_j) = (\pi_1, \dots, \pi_k)$ in which the π_i 's are computed from the iterative scheme:

$$\pi_i^{(n)} = \pi_i^{(n-1)} \left(\sum_{r=1}^m \sum_{\ell=1}^{g_r} (f_{r\ell}^{(j)} q_{ir\ell} / \sum_{u=1}^k \pi_u^{(n-1)} q_{ur\ell}) \right) / m$$

for $i = 1, \dots, k$.

Where the data are crisp, i.e. if we are trying to estimate π_i from a singleton partition set $S_{ri}^{(j)}$ of tuple r , then the contribution to π_i is simply $f_{ri}^{(j)}$. Otherwise we apportion the probability $f_{ri}^{(j)}$ to the π_b 's where $v_b \in S_{bi}^{(j)}$ in the proportion $\pi_i / (\sum_{\gamma \in S_{bi}^{(j)}} \pi_\gamma)$. We illustrate this idea in the

following example.

Example 2: Consider the first two tuples of the attribute HEART_DISEASE in Table 1.

Here $g_1 = 2$; $S_{11} = \{S\}$; $S_{12} = \{M, N\}$; $f_{11} = 0.75$; $f_{12} = 0.25$; $q_{111} = 1$; $q_{212} = 1$; $q_{312} = 1$;

and $g_2 = 3$; $S_{21} = \{S\}$; $S_{22} = \{M\}$, $S_{23} = \{N\}$; $f_{21} = 0.75$; $f_{22} = 0.2$; $f_{23} = 0.05$; $q_{121} = 1$; $q_{222} = 1$; $q_{323} = 1$.

The iteration scheme in Definition 4 is then given by:

$$\pi_1^{(n)} = \pi_1^{(n-1)} (0.75 / \pi_1^{(n-1)} + 0.75 / \pi_1^{(n-1)}) / 2$$

$$\pi_2^{(n)} = \pi_2^{(n-1)} (0.25 / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) + 0.2 / \pi_2^{(n-1)}) / 2$$

$$\pi_3^{(n)} = \pi_3^{(n-1)} (0.25 / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) + 0.05 / \pi_3^{(n-1)}) / 2$$

from which we obtain the result that:

$$\pi_1 = 0.75$$

$$\pi_2^{(n)} = 0.125 \pi_2^{(n-1)} / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) + 0.1$$

$$\pi_3^{(n)} = 0.125 \pi_3^{(n-1)} / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) + 0.025.$$

By iteration, or simply substitution, we obtain the result that $\pi_1 = 0.75$, $\pi_2 = 0.2$, $\pi_3 = 0.05$.

Our definition of the general aggregate operator provides a general approach which can aggregate both imprecise and uncertain data in a consistent manner. The way the algorithm treats imprecise data is intuitively appealing since imprecise data values are apportioned to the appropriate aggregated probabilities according to the probabilities of their taking each of the possible values. We now show that the general aggregation algorithm also has a strong theoretical underpinning.

The Theoretical Framework

The *Kullback-Leibler information divergence* between two distributions $p = (p_1, \dots, p_n)$ and $q = (q_1, \dots, q_n)$ is defined as

$$D(p \parallel q) = \sum_j p_j \log(p_j / q_j).$$

The iteration equations which define the aggregation in fact minimise the Kullback-Leibler information divergence between the aggregated probability distribution $\{\pi_i\}$ and the data $\{f_{rs}\}$. This is equivalent to maximising

the likelihood of the model given the data. Minimising the Kullback-Leibler information divergence, or equivalently maximising the log-likelihood, becomes:

$$\text{Maximise } W = \sum_{r=1}^m \sum_{\ell=1}^{t_r} f_{r\ell}^{(j)} \log \left(\sum_{i=1}^k q_{ir\ell} \pi_i \right)$$

$$\text{subject to } \sum_{i=1}^k \pi_i = 1.$$

Vardi and Lee (1993) have shown that this problem belongs to a general class which they term Linear Inverse Problems (LININPOS). For such problems they develop an algorithm which converges monotonically to the solution of the minimum information divergence equation. In our case this algorithm is equivalent to Definition 4, and is an example of the EM (expectation-maximisation) algorithm (Dempster et al, 1977), which is widely used for the solution of incomplete data problems. For example, such an approach is used by Scotney and McClean (1998) to integrate distributed data over attribute domain partitions. In general terms our problem may be regarded as one of graphical modelling, as described by Cox and Wermuth (1996), in this case subject to statistical incompleteness.

Vardi and Lee (1993) have shown that the iterative scheme always converges, but where there is an identifiability problem the solution may not be unique. However, they show that a sufficient condition for uniqueness is that the rows of the matrix $\mathbf{Q} = \{q_{i\ell}\}$ are linearly independent. They also show that a necessary and sufficient condition for uniqueness is that the equation $\mathbf{F} = \boldsymbol{\pi} \mathbf{Q}$ does not have multiple solutions, where $\mathbf{Q} = \{q_{i\ell}\}$ and $\boldsymbol{\pi} = \{\pi_i\}$. The vector \mathbf{F} consists of elements which are the sums of the probabilities f_{rs} aggregated over the set S_{rs} . The vector is then normalised so that the elements sum to 1. In our case, since it is likely that we will have a large proportion of the data which is crisp, this condition will normally be satisfied.

In general, convergence of the EM algorithm may be slow (Wu, 1983). However, if most values are crisp, it is likely that the rate of convergence will not be a problem. Our simulation results support this claim. However, if speed of convergence becomes a problem, then there are a number of recent papers which suggest ways in which convergence can be accelerated, e.g. Jamshidian and Jennrich (1997), Molenberghs and Goetghebeur (1997).

In our simulations we have used the scaled cardinalities of all the crisp data as an initial value for the general aggregation operator. This is the answer we would get if we applied the simple aggregation operator and ignored all imprecise data. Such a starting value should therefore prove reasonably close to the final answer and therefore lead to fast convergence.

We now illustrate the approach by applying the algorithm to the attribute HEART_DISEASE in Table 1.

Example 3: In this case the matrix \mathbf{Q} is given by

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

corresponding to $\{S\}$, $\{M\}$, $\{N\}$, and $\{M, N\}$ respectively, i.e. we have permuted the rows and columns to reflect a logical ordering of the subsets and have deleted repeated columns. The vector \mathbf{F} , defined earlier, is then given by:

$$\mathbf{F} = (0.49, 0.145, 0.18, 0.185).$$

The iterative scheme for this data is thus:

$$\begin{aligned} \pi_1^{(n)} &= 0.49 \\ \pi_2^{(n)} &= 0.145 + 0.185 * \pi_2^{(n-1)} / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) \\ \pi_3^{(n)} &= 0.18 + 0.185 * \pi_3^{(n-1)} / (\pi_2^{(n-1)} + \pi_3^{(n-1)}), \end{aligned}$$

from which we obtained the result that:
 $\pi_1 = 0.49$, $\pi_2 = 0.228$, $\pi_3 = 0.282$
in one iteration (to 3 decimal places), using a start value of (0.601, 0.178, 0.221).

Aggregation involving Several Attributes

In the previous section we illustrated the use of a general aggregation operator for imprecise and uncertain data in a single attribute. However, it is often of more interest to derive the joint probabilities of several attributes' values occurring simultaneously, and our methodology may be extended to cover this situation. Hence we consider the cross product of attributes A_1, \dots, A_n with domains D_1, \dots, D_n . We define the cross product $\mathcal{A} = A_1 \times \dots \times A_n$ with domain $\mathcal{D} = D_1 \times \dots \times D_n$. Let the values of domain D_j be given by $\{v_1^{(j)}, \dots, v_{k_j}^{(j)}\}$. Then the values of \mathcal{D} are of the form $\{v_{a_1}^{(1)} \times \dots \times v_{a_n}^{(n)}\}$ where $a_i \in \{1, \dots, k_i\}$ for $i = 1, \dots, n$.

The partial probability distribution then assigns probabilities $\{f_{rs}\}$ to partitions $\{S_{rs}\}$ of \mathcal{D} for each tuple. The algorithm proceeds as in Definition 4. In Example 4 we illustrate this approach by deriving the joint distribution of the attributes SMOKING_STATUS and HEART_DISEASE in Table 1.

Example 4: The possible singleton values of the domain $\mathcal{D} = \text{SMOKING_STATUS} \times \text{HEART_DISEASE}$ are: $\{S, S\}$, $\{S, M\}$, $\{S, N\}$, $\{E, S\}$, $\{E, M\}$, $\{E, N\}$, $\{N, S\}$, $\{N, M\}$, $\{N, N\}$ with corresponding probabilities π_1, \dots, π_9 . The subsets of \mathcal{D} which are represented in the data are: $\{S, S\}$, $\{S, M\}$, $\{S, N\}$, $\{E, S\}$, $\{E, M\}$, $\{E, N\}$, $\{N, S\}$, $\{N, M\}$, $\{N, N\}$, $\{S, \{M, N\}\}$, $\{E, \{M, N\}\}$, and for which the vector \mathbf{F} is

$$\mathbf{F} = \{0.375, 0.08, 0.005, 0.105, 0.045, 0.025, 0.01, 0.02, 0.15, 0.105, 0.08\}.$$

The iterative scheme for this data is given by:

$$\begin{aligned} \pi_1^{(n)} &= 0.375 \\ \pi_2^{(n)} &= 0.08 + 0.105 * \pi_2^{(n-1)} / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) \\ \pi_3^{(n)} &= 0.005 + 0.105 * \pi_3^{(n-1)} / (\pi_2^{(n-1)} + \pi_3^{(n-1)}) \\ \pi_4^{(n)} &= 0.105 \\ \pi_5^{(n)} &= 0.045 + 0.08 * \pi_5^{(n-1)} / (\pi_5^{(n-1)} + \pi_6^{(n-1)}) \\ \pi_6^{(n)} &= 0.025 + 0.08 * \pi_6^{(n-1)} / (\pi_5^{(n-1)} + \pi_6^{(n-1)}) \\ \pi_7^{(n)} &= 0.01 \\ \pi_8^{(n)} &= 0.02 \\ \pi_9^{(n)} &= 0.15 \end{aligned}$$

from which we obtain the result that:

$$\begin{aligned} \pi_1 &= 0.375, \pi_2 = 0.179, \pi_3 = 0.011, \pi_4 = 0.105, \\ \pi_5 &= 0.096, \pi_6 = 0.054, \\ \pi_7 &= 0.01, \pi_8 = 0.02, \pi_9 = 0.15, \end{aligned}$$

where convergence occurred in one iteration.

Knowledge Discovery

In order to discover knowledge from databases we are often concerned with inducing rules from database tuples (Anand et al., 1997). Rules are often based on sets of attribute values partitioned into an antecedent and a consequent. A typical "if then" rule of the form "if antecedent = true, then consequent = true" is given by "if an individual smokes and is hypertensive, then he or she is very likely to suffer from heart disease". Support for such a rule is based on the proportion of tuples in the database that have the specified attribute values in both the antecedent and the consequent. Therefore the evaluation of such an association rule often requires only standard database functionality, and may be implemented using embedded SQL (Anand et al., 1997).

Using the approach described in previous sections we may use the general aggregation operator to derive joint probabilities and then compute conditional probabilities. Thus, for example, using the data in Table 1 we obtain:

$$\begin{aligned} \text{probability(smokes and severe heart disease)} &= 0.375 \\ \text{probability(smokes)} &= 0.565. \end{aligned}$$

Therefore, the conditional probability of having severe heart disease given that an individual smokes is given by:

$$\begin{aligned} \text{probability(severe heart disease | smokes)} \\ = 0.375 / 0.565 = 0.664. \end{aligned}$$

However, the unconditional probability (severe heart disease) = 0.49, which is smaller. We may therefore possibly induce the rule "smoking increases the chances of heart disease", subject to thresholding parameters.

We may therefore use the general aggregation operator to assess rule validity. Thus, in general, if we wish to examine the rule:

$$A_1, \dots, A_r \rightarrow B_1, \dots, B_s,$$

where $A_1, \dots, A_r, B_1, \dots, B_s$ are all attributes of a relation R , then we compute the joint probabilities:

$$\text{Prob}(A_1, \dots, A_r, B_1, \dots, B_s) \text{ and } \text{Prob}(A_1, \dots, A_r),$$

from which we may assess the validity of the rule by calculating the conditional probability:

$$\text{Prob}(B_1, \dots, B_s | A_1, \dots, A_r) = \text{Prob}(A_1, \dots, A_r, B_1, \dots, B_s) / \text{Prob}(A_1, \dots, A_r).$$

We may thus determine the strength and support for various rules under consideration. Thresholding then takes place to decide if the candidate rules are sufficiently likely to merit the user being informed. A probabilistic approach, such as that described, may then be used to represent rules as linguistic summaries (McClellan and Scotney, 1997) in order to facilitate a user interface to the KDD process.

Summary and Further Work

We have considered an extended relational data model which handles both imprecise and uncertain data. We have also developed general aggregation operator which allows us to determine a probability distribution for attribute values either for a single attribute or for the cross-product of a number of attributes. Such functionality has potential for knowledge discovery in databases where we may use it to assess the validity of association rules. Further work will focus on including domain knowledge, such as that held as integrity constraints, into the model, and on techniques for accelerating the algorithm.

Acknowledgements

This work was partially funded by IDARESA (ESPRIT project no. 20478) and partially funded by ADDSIA (ESPRIT project no. 22950) which are both part of EUROSTAT's DOSIS (Development of Statistical Information Systems) initiative.

References

- Anand S.S., Scotney B.W., Tan M.G., McClellan S.I., Bell D.A., Hughes J.G., and I.C. Magill (1997). Designing a kernel for data mining. *IEEE Expert*, **12** (2), 65 - 74.
- Barbará D., Garcia-Molina H. and D. Porter (1992). The management of probabilistic data. *IEEE Transactions on Knowledge and Data Engineering*, **4**, 487-501.
- Chang C.-S. and A.L.P. Chen (1994). Determining probabilities for probabilistic partial values. *Proc. Int. Conf. Data and Knowledge Systems for Manufacturing & Engineering*, Hong Kong, 277-284.
- Chen A.L.P. and F.S.C. Tseng (1996). Evaluating aggregate operations over imprecise data. *IEEE Transactions on Knowledge and Data Engineering*, **8**, 273-284.
- Cox D.R. and N. Wermuth (1996). *Multivariate Dependencies - Models, analysis and interpretation*. Chapman and Hall.
- DeMichiel L.G. (1989). Resolving database incompatibility: an approach to performing relational operations over mismatched domains. *IEEE Transactions on Knowledge and Data Engineering*, **4**, 485-493.
- Dempster A.P., Laird N.M. and D.B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Statist. Soc. B*, **39**, 1-38.
- Jamshidian J.M. and R.I. Jennrich (1997). Acceleration of the EM algorithm by using quasi-Newton methods. *J. R. Statist. Soc. B*, **59**, 569-587.
- McClellan S.I., and B.W. Scotney (1997). Using evidence theory for the integration of distributed databases. *International Journal of Intelligent Systems*, **12** (10), 763-776.
- Molenberghs G. and E. Goetghebeur (1997). Simple fitting algorithms for incomplete categorical data. *J. R. Statist. Soc. B*, **59**, 401-414.
- Parsons S. (1996). Current approaches to handling imperfect information in data and knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, **8**, 353-372.
- Scotney B.W., McClellan S.I., and M.C. Rodgers (1998). Optimal and efficient integration of heterogeneous data in a distributed database. *Data and Knowledge Engineering Journal*, forthcoming.
- Tseng F.S.C., Chen A.L.P. and W.-P. Yang (1993). Answering heterogeneous database queries with degrees of uncertainty. *Distributed and Parallel Databases*, **1**, 281-302.
- Vardi Y. and D. Lee (1993). From image deblurring to optimal investments: maximum likelihood solutions for positive linear inverse problems (with discussion). *J. R. Statist. Soc. B*, 569-612.
- Wu C.F.J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, **11** (1), 95-103.