Crowd-sourced Targeted Feedback Collection for Multi-Criteria Data Source Selection

JULIO CÉSAR CORTÉS RÍOS, NORMAN W. PATON, ALVARO A.A. FERNANDES, EDWARD ABEL, and JOHN A. KEANE, University of Manchester, United Kingdom

A multi-criteria data source selection (MCSS) scenario identifies, from a set of candidate data sources, the subset that best meets users needs. These needs are expressed using several criteria, which are used to evaluate the candidate data sources. A MCSS problem can be solved using multi-dimensional optimisation techniques that trade-off the different objectives. Sometimes one may have uncertain knowledge regarding how well the candidate data sources meet the criteria. In order to overcome this uncertainty, one may rely on end users or crowds to annotate the data items produced by the sources in relation to the selection criteria. In this paper, a proposed Targeted Feedback Collection (TFC) approach is introduced, that aims to identify those data items on which feedback should be collected, thereby providing evidence on how the sources satisfy the required criteria. The proposed TFC targets feedback by considering the confidence intervals around the estimated criteria values, with a view to increasing the confidence in the estimates that are most relevant to the multi-dimensional optimisation. Variants of the proposed TFC approach have been developed, for use where feedback is expected to be reliable (e.g. where it is provided by trusted experts) and where feedback is expected to be unreliable (e.g. from crowd workers). Both variants have been evaluated, and positive results are reported against other approaches to feedback collection, including active learning, in experiments that involve real world data sets and crowdsourcing.

CCS Concepts: • **Information systems → Mediators and data integration**;

Additional Key Words and Phrases: source selection, feedback collection, pay-as-you-go, single-objective optimisation, multi-objective optimisation, crowd-sourcing, uncertainty handling

ACM Reference Format:
Julio César Cortés Ríos, Norman W. Paton, Alvaro A.A. Fernandes, Edward Abel, and John A. Keane. 2018. Crowd-sourced Targeted Feedback Collection for Multi-Criteria Data Source Selection. *ACM J. Data Inform. Quality* 1, 1, Article 1 (January 2018), 25 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The number of available data sources is increasing at an unprecedented rate [Halevy et al. 2016]. Open data initiatives and other technological advances, like publishing to the web of data or automatically extracting data from tables and web forms, are making the source selection problem a critical topic. In this context, it is crucial to select those data sources that satisfy user requirements on the basis of well-founded decisions.

Regarding the properties that the data sources must exhibit, there have been studies of the data source selection problem considering specific criteria, such as accuracy, cost and freshness [Dong et al. 2012; Rekatsinas et al. 2014].

Authors' address: Julio César Cortés Ríos, juliocesar.cortesrios@manchester.ac.uk; Norman W. Paton; Alvaro A.A. Fernandes; Edward Abel; John A. Keane, University of Manchester, School of Computer Science, Oxford Road, Manchester, M13 9PL, United Kingdom.

Manuscript submitted to ACM

This paper adopts a multi-criteria approach that can be applied to diverse criteria in order to accommodate a wider variety of user requirements and preferences, while considering the trade-off between the required criteria. In this approach, from a collection of data sources, S, the problem is to identify a subset of the data sources S' from which R data items can be obtained that reflect users preferences. These preferences are represented by a collection of weighted criteria; for example, the criteria could be of the form *accuracy:0.4, freshness:0.3, relevance:0.3*, indicating that *freshness* and *relevance* are of equal importance to the user, and that *accuracy* is more important still.

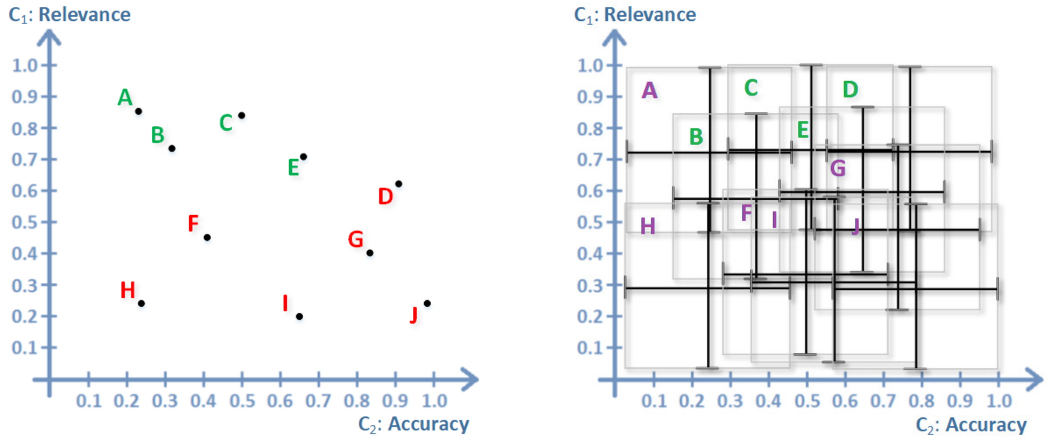
To solve the multi-criteria data source selection (MCSS) problem, a multi-dimensional optimisation technique is used to provide a solution that takes into account users preferences (represented as weights) in relation to the criteria [Abel et al. 2018]. This paper addresses the MCSS problem using an approach where the objective is to retrieve an optimal number of items from each data source given the weighted criteria that model the users requirements.

To inform the data source selection process, where criteria estimates are likely to be unreliable, one needs to annotate the candidate data sources to obtain their criteria values; this is an essential step, as one needs to know how each data source scores for each criterion. Given that there may be many data sources and criteria, the data source annotation process can become expensive. In this paper, the focus is on a pay-as-you-go approach, that collects feedback in the form of true and false positive annotations on data items, that indicate whether or not these data items satisfy a specific criterion. Such feedback could come from end users or crowd workers, and has been obtained in previous works ([Belhajjame et al. 2013; Bozzon et al. 2012; Franklin et al. 2011; Ríos et al. 2016]).

Having an efficient way to collect the feedback required to improve the knowledge about the data sources is important, as there are costs involved. Hence, there is the need to carefully identify the data items on which to ask for feedback in order to maximise the effect of the new evidence collected, and to minimise the amount of feedback that needs to be collected. Some recent work has focused on targeting feedback, especially in *crowdsourcing* (e.g. [Crescenzi et al. 2017; Li et al. 2016]); here such work is complemented by providing an approach to feedback collection for multi-criteria data source selection. Although the impact of uncertainty in data integration tasks has been explored in previous works [Magnani and Montesi 2010], the proposed TFC approach differs from other approaches to feedback targeting in building on confidence intervals on criterion estimates. In so doing, the proposed approach addresses two kinds of uncertainty: uncertain criterion estimates and unreliable feedback.

The importance of using feedback to improve information processing to provide better user experience extends beyond the data source selection problem addressed in this research and it has been applied to several real-world problems. For instance, in image retrieval, relevance feedback has been widely used in previous works [Barbu et al. 2013; Gallas et al. 2014; Villegas et al. 2013] to raise the quality of the results in terms of how closely they reflect the user's intention. Implicit and explicit user feedback has also been used to improve neural to machine translations [Kreutzer et al. 2018], to amend the detection of bugs during automated application testing [Grano et al. 2018], and to enhance the interaction between adaptive robots and humans [Muthugala and Jayasekara 2017]. Hence, the search of mechanisms to make the feedback collection process cost-effective, such as the proposed TFC approach described in this paper, is becoming increasingly relevant.

To provide a practical example of the problems addressed in this paper, and how the proposed TFC approach is used to improve the solution to the MCSS problem, consider the example presented in Figure 1 (a). In this scenario, a user is looking for nutritional information on popcorn products and there are currently ten different sources that provide this information. This ten data sources are labelled $A, B, C, D, E, F, G, H, I, J$, each with different values in terms of their relevance (related to nutritional information about popcorn products) and accuracy (the correctness of the nutritional information presented) to the search. The MCSS problem for this example is which combination of those ten



(a)

(b)

Fig. 1. Multi-criteria data source selection without (a) and with (b) uncertainty.

data sources maximises the relevance and the accuracy of the results to the user inquiry. This is already a non-trivial problem that becomes more complex as the number of criteria increases. But, what if the values available for each data source over each of the criteria in contention are incomplete or unknown? Then the previous scenario can be depicted as in Figure 1 (b), where the original values are now surrounded by margins of error that represent the uncertainty for those values, making the selection problem even harder. To reduce this uncertainty one can rely on feedback to refine the estimated criteria values, hence, the proposed TFC approach is designed to aid in the identification of the feedback required to reduce this uncertainty and improve the MCSS solutions.

The following contributions are reported in this paper:

(1) a strategy for targeted feedback collection for use with MCSS with an arbitrary number of weighted criteria;
(2) an algorithm that implements the strategy, using the confidence intervals around the criteria estimates to identify those sources that require more feedback to improve the results of MCSS;
(3) an extension to (1) and (2) that accommodates unreliable feedback; and
(4) an experimental assessment of the proposed TFC approach using real-world data to show that TFC can consistently and considerably reduce the amount of feedback required to achieve high-quality solutions for the MCSS problem compared to the proposed baseline (random selection of data items) and an active learning-based technique called uncertainty sampling [Lewis and Gale 1994].

This paper is an extended version of [Ríos et al. 2017]; the extensions in this paper include additional experiments on the algorithm for reliable feedback (2), support for unreliable feedback (3), experiments with crowd workers whose feedback can be expected to be unreliable, and an extended discussion of related work.

Overall, this paper can be considered to be contributing on research into data quality: by supporting data source selection using criteria that capture the fitness for purpose of the data; by enabling data source selection to be informed by a wide range of quality criteria; by targeting feedback in ways that improves the quality of the criteria estimates; and by taking into account the reliability of the feedback in targeting feedback collection.

This paper is structured as follows. Section 2 provides a detailed definition of the problem. The concepts on which TFC builds are described in Section 3. Section 4 presents the TFC strategy and algorithm for targeting reliable feedback.

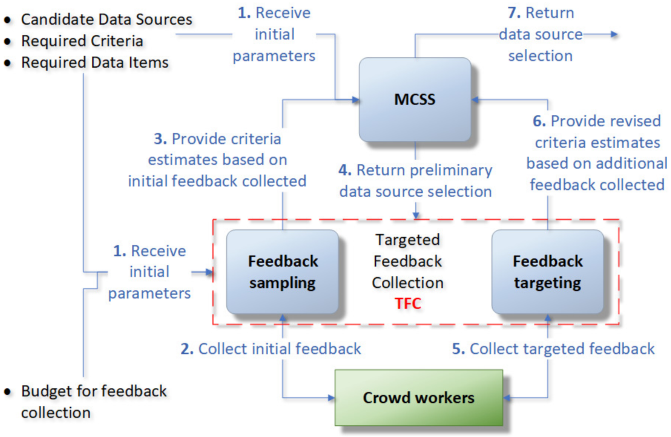

• Candidate Data Sources
• Required Criteria
• Required Data Items
1. Receive initial parameters
7. Return data source selection
MCSS
3. Provide criteria estimates based on initial feedback collected
4. Return preliminary data source selection
6. Provide revised criteria estimates based on additional feedback collected
1. Receive initial parameters
Feedback sampling
Targeted Feedback Collection
TFC
Feedback targeting
2. Collect initial feedback
5. Collect targeted feedback
• Budget for feedback collection
Crowd workers

Fig. 2. Relation between TFC and MCSS

The experimental results comparing TFC against random and uncertainty sampling are in Section 5. The extension to accommodate unreliable feedback is described in Section 6, and evaluated with crowd workers using Crowdflower (now Figure Eight) [1] in Section 7. Related work is discussed in Section 8, and conclusions are given in Section 9.

2 PROBLEM DESCRIPTION

MCSS is a complex problem when the number of criteria is large and the user can declare preferences over these criteria. Concretely, the MCSS problem can be defined as: given a set of candidate data sources $S = \{s_1, \ldots, s_m\}$, a set of user criteria $C = \{c_1, \ldots, c_n\}$ with weights $W = \{w_1, \ldots, w_n\}$, and a target number of data items R, identify a collection $\mathbf{X}$ with m elements, indicating how many data items each data source in S contributes to the solution. The data sources $S' = \{s_i | s_i \in S, \mathbf{X}[i] > 0\}$ contribute to the solution, and those in $S \setminus S'$ do not.

The relation between the proposed targeted feedback collection approach (TFC) and the multi-criteria data source selection (MCSS) problem is presented in Figure 2, showing the interaction between the MCSS problem and the proposed TFC approach to target the feedback required to improve the criteria estimates of the candidate data sources. The first step involves the preparation of the scenario by considering the candidate data sources, the criteria to evaluate those data sources and the number of required data items, additionally the budget for feedback collection is allocated for the TFC approach. Secondly, TFC collects feedback on a random sample over the data items produced by all the candidate data sources. This feedback is used in the third step to compute the estimated criteria values that are used to solve the MCSS problem. In the fourth step, the data sources contributing to the solution to the MCSS problem are taken into account by the TFC strategy to target additional data items on which to collect feedback (fifth step), in order to refine the criteria estimates that are iteratively used to solve the MCSS problem (sixth step). Steps 4, 5 and 6 are repeated until the solution to the data source selection problem can not be improved by refining the criteria estimates, or the budget allocated for feedback collection is exhausted. The final MCSS solution is returned in step 7.

In this research, two scenarios were considered to provide the feedback required to improve the criteria estimates. The first one is based on reliable feedback assumed to be from expert users and the second relies on a crowdsourcing platform where workers provide the feedback. In the first case, the feedback can be considered as completely reliable

[1] https://www.figure-eight.com/

and, hence, it does not provide additional uncertainty to the selection problem; the TFC approach for this scenario is presented in Section 4 and evaluated in Section 5. For the second scenario, the crowdsourced workers introduce additional uncertainty, as they may provide incorrect answers by accident or on purpose. Hence, to consider this unreliable feedback, an improved TFC approach is described in Section 6 and evaluated in Section 7.

To describe the problems addressed in this paper in detail consider again the practical example presented in Section 1 about the selection of data sources in the food facts domain as presented in Figure 1 (a) with the ten data sources $S = \{A, B, C, D, E, F, G, H, I, J\}$, and two data property criteria to balance: relevance (c_1) and accuracy (c_2) with the following weights $W = \{w_1 = 0.5, w_2 = 0.5\}$. The user requires a particular number of data items (R) from a subset of data sources in S that maximise both criteria and reflect the weights W (in this case, the user considers the two criteria to be of identical importance). The first criterion, in this case, evaluates how relevant a data item is according to the user's expectations, for example, which items are relevant if looking for popcorn products? The second criterion evaluates the accuracy of the attribute values on each data item; e.g., a data item would be considered accurate only if the nutritional information is correctly stated and consistent with each product.

This problem can be solved by using linear programming or other multi-dimensional optimisation techniques [Abel et al. 2018]. For this research, an additional factor is being considered, the presence of uncertainty in the data source criteria estimates. This uncertainty is caused by incomplete evidence, and can be reduced by annotating data items produced by each data source to determine if they satisfy the conditions for each criterion. The proposed TFC strategy identifies the data items that reduce this uncertainty to support better solutions for the MCSS problem.

To solve the MCSS problem an optimisation technique can be applied, such as that presented in Subsection 3.3, to obtain a solution $\mathbf{X}$ which is a vector containing the values of the decision variables x after the optimisation for all the sources. This solution indicates how many data items each data source from S contributes. In Figure 1 (a), assume that the solution includes data items from the data sources in $S' = \{A, B, C, E\}$, in other words, the data sources in S' contribute data items to the solution.

This paper addresses the case where there is uncertain knowledge about the criteria values, further complicating the problem. In Figure 1 (b), instead of dots representing the data sources in the multi-criteria space, there are bi-dimensional intervals representing the uncertainty around each data source's criteria values. The real criteria values may be expected to lie within this area, but, in the absence of more evidence, one does not know exactly where. For instance, if the MCSS is solved against the estimated criteria values of data sources in S as shown in Figure 1 (b), then, a solution containing a different subset of data sources, viz., $\{B, C, D, E\}$, may be obtained due to the uncertainty associated with each estimated criterion, hence, leading to poor solutions to the MCSS problem. Therefore, now the question is: how can one cost-effectively select data items on which to collect feedback, in order to reduce the uncertainty in a way that benefits the optimisation technique in solving the MCSS problem? The proposed TFC is an approach that minimises the number of data items on which feedback needs to be collected, and determines the point beyond which the collection of more feedback can be expected to have no effect on which data sources are selected, as presented in Figure 2.

3 TECHNICAL BACKGROUND

To model different criteria under a common framework data quality techniques are used. Multi-dimensional optimisation techniques, particularly for integer linear functions, are used to find a solution for the MCSS problem. The dimensions in this case represent criteria that need to be balanced to find the best solution. These concepts are defined in this section and are used to describe the proposed TFC approach in Section 4.

3.1 Data Criteria

A data criterion is a metric that can be applied to a data set to evaluate how the data set fares on that criterion. There are many different criteria that can be applied to the data source selection problem. For instance, in [Pipino et al. 2002; Rekatsinas et al. 2015] accuracy (degree of correctness) and freshness (how recent is the information) were used. In [Ríos et al. 2016], data sources are evaluated based on their precision and recall.

In this paper, the estimated value of a data criterion $\hat{c}$ is evaluated as the ratio between the elements satisfying the notion for a given metric (and for which feedback has been collected) or true positives, tp, and all the annotated elements, i.e., the sum of the true and false positives, fp. For example, to evaluate the relevance of a data source, the number of relevant data items is divided over the total number of data items labelled for that source. The following formula calculates $\hat{c}$ using the data items on which feedback has been collected for a data source s:

$$\hat{c}_s = \frac{|tp_s|}{|tp_s| + |fp_s|} \tag{1}$$

For instance, the definition of relevance could be: *a data item from the food facts domain is considered to be relevant if and only if it is associated with popcorn products.*

3.2 Confidence Interval, Overlap and Sample Size

The proposed TFC strategy takes into account the data sources that should contribute to a solution (given a collection of data criteria) and those that should not. This is done by analysing the overlapping of the confidence intervals around the criteria value estimates for each data source. A *confidence interval* (CI) is the range formed by flanking the estimated value (based on the available evidence or feedback) with a margin of error for a required confidence level, and represents the space in which the true value is expected to be contained. This CI is associated with a given *confidence level*, which is the percentage of all possible samples that can be expected to include the real value. Following this definition, the larger the number of data items labelled for a data source, the greater the confidence in the estimated values, and hence, the smaller the CIs around these estimates. The following formulae are used to compute the CIs for a data source s [Bulmer 1979] (assuming the data is normally distributed as the real distribution is unknown):

$$e_s = z_{cL} \cdot se_s \cdot fpc_s = z_{cL} \cdot \sqrt{\frac{\hat{c}_s \cdot (1 - \hat{c}_s)}{L_s}} \cdot \sqrt{\frac{T_s - L_s}{T_s - 1}} \tag{2}$$

To compute the upper and lower bounds of the CI the following equations are used:

$$upCI = \min(\hat{c}_s + e_s, 1.0) \tag{3}$$

$$lowCI = \max(\hat{c}_s - e_s, 0.0) \tag{4}$$

where s is a source in the set of candidate data sources S, se_s is the *standard error*, fpc_s is the *finite population correction factor* used to accommodate data sources of any size assuming that they have a finite number of data items, L_s is the number of feedback instances collected for s, T_s is the total number of data items produced by s, $\hat{c_s}$ is the estimated data criterion, and $lowCI$ and $upCI$ are the lower and upper bounds of the CI, respectively. The result is the *margin of error* e_s around the estimate, e.g. $\hat{c}_s \pm e_s$, for a given confidence level cL and its corresponding z–score z_{cL}.

The proposed TFC strategy relies on the CIs surrounding the criteria estimates for each data source, and how these CIs overlap. The approach [Knezevic 2008] is to determine not only if two CIs overlap, but also the amount of overlap. Analysis of this overlap determines whether the two CIs overlap or not and, if so, whether their means are significantly

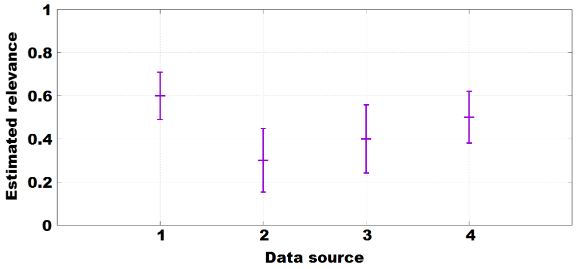


Fig. 3. Overlapping of confidence intervals

different or not. If the means of the overlapping intervals are not significantly different, then, either both remain in contention to be part of a solution or both are discarded, and, for TFC, both are considered to be *significantly overlapping*.

Assuming that the estimated value $\hat{c}_{s_1}$ is higher than $\hat{c}_{s_2}$, then, there is *overlap* between the CIs if:

$$\hat{c}_{s_1} - e_{\hat{c}_{s_1}} \leq \hat{c}_{s_2} + e_{\hat{c}_{s_2}} \text{ or } \hat{c}_{s_1} - \hat{c}_{s_2} \leq (e_{\hat{c}_{s_1}} + e_{\hat{c}_{s_2}}) \tag{5}$$

The means of the CIs of s_1 and s_2 are *not significantly different*, hence, the intervals are *significantly overlapping*, if:

$$\hat{c}_{s_1} - \hat{c}_{s_2} \leq \sqrt{e^2_{\hat{c}_{s_1}} + e^2_{\hat{c}_{s_2}}} \tag{6}$$

For instance, assume four data sources with initial estimated relevance, in regards to a user query, and CIs as shown in Figure 3. In this plot, only s_1 contributes to the solution to the data source selection problem, thus, the overlap is analysed between the CI of s_1 and the CIs of the other sources. Using Equations 5 and 6, the following results:

(1) There is *no overlap* between the CIs of s_1 and s_2 as the lower bound of the CI of s_1 is greater than the upper bound of the CI of s_2.
(2) There is *overlap* between the CIs of s_1 and s_3 but the *difference between the means is significant.*
(3) There is *overlap* between the CIs of s_1 and s_4 and the *difference between the means is not significant.*

Based on the previous observations and following the TFC strategy, the result of the analysis is that only s_1 and s_4 must be considered for additional feedback collection. s_1 because it is already contributing to the solution, and s_4 because its CI significantly overlaps with the CI of s_1.

The TFC strategy uses the *sample size sS* for a finite population T [Foley 1972; Lewis and Sauro 2006] to compute the number of data items required to have a representative sample of the entire population given a confidence level and a margin of error. Feedback on the sampled elements is then collected to compute an initial estimate of the underlying data quality. sS is also used to estimate the number of elements required during each feedback collection episode:

$$sS_T = \frac{\frac{z^2_{cL} \cdot (\hat{c}_s \cdot (1-\hat{c}_s))}{e^2}}{1 + \frac{\frac{z^2_{cL} \cdot (\hat{c}_s \cdot (1-\hat{c}_s))}{e^2} - 1}{|T|}} \tag{7}$$

3.3 Multi-Criteria Optimisation

Considering that the data criteria are evaluated using linear functions, linear-programming techniques can be used to find the optimal solution that balances all the criteria and user preferences (represented as weights). To solve the optimisation problem, a Multi-objective linear programming (MOLP) approach called *Min-sum* has been selected [Abel et al. 2018]. This approach seeks to minimise the summed weighted deviation from the ideal solution across the multiple dimensions. Min-sum has been selected due to its simplicity in trading off the different criteria and low computational requirements, and can be represented as a linear programming model with a collection of objective functions **Z** and their associated constraints. The solution is a vector **X** with the values of the decision variables x after the optimisation.

Following the Min-sum approach, the first step is to obtain the ideal Z_k^* and negative ideal Z_k^{**} solutions (best and worst possible solutions, respectively) for each criterion k by using single objective optimisation. These solutions are found by optimising each criterion with respect to the following single objective function Z:

$$Z_k = \frac{\sum_{i=1}^{m} x_i \cdot \hat{c}_{k_{s_i}}}{R} \quad k = 1, 2, \ldots, n, \tag{8}$$

where m is the number of data sources available, n is the number of user criteria, x_i is the number of data items used from data source s_i, $\hat{c}_{k_{s_i}}$ is the value of the criterion k for s_i, and R is the number of data items requested.

Z_k is solved as both maximisation and minimisation objective functions with the following constraints. The number of data items chosen from each source in S, x_i, cannot exceed the number of items $|s|$ produced by s:

$$x_i \leq |s_i| \quad i = 1, 2, \ldots, m \tag{9}$$

The total number of data items chosen must equal the amount requested:

$$\sum_{i=1}^{m} x_i = R \tag{10}$$

And the minimum value for the decision variables x is 0 (non-negativity):

$$x_i \geq 0 \quad i = 1, 2, \ldots, m \tag{11}$$

The ideal and negative ideal solutions for each criterion k are then computed to obtain the range of possible values. These solutions, along with the constraints from Equations 9-11, and the user preference weights w are used to find a solution that minimises the sum of the criteria deviations. Each per-criterion deviation measures the compromise in a solution with respect to the corresponding ideal value given the user weights.

The weighted deviation for each criterion D_k shows how far the current solution is from the ideal:

$$D_k = \frac{w_k \cdot (Z_k^* - Z_k)}{Z_k^* - Z_k^{**}} \quad k = 1, 2, \ldots, n, \tag{12}$$

And finally, the optimisation model consists in minimising the sum of criteria deviations λ (measure of the overall deviation from the objective) by considering the constraints in Equations 9-12 as follows:

$$\min \quad \lambda,$$

where:

$$\lambda = D_1 + D_2 + \ldots + D_n, \tag{13}$$

4 TARGETING FEEDBACK USING MULTI-DIMENSIONAL CONFIDENCE INTERVALS

4.1 Overview

In this section, the TFC strategy is defined following the practical problem described in Sections 1 and 2, where the goal is to select, from the available candidates, the data sources that provide the maximum combined relevance and accuracy to support an analysis of nutritional information of popcorn products. For this goal, some budget was allocated to fund feedback on b data items. The assumption is that there is no up-front knowledge of the relevance or accuracy.

Following Figure 2, initial estimates about the values of the criteria for the candidate data sources are needed. These initial estimates are obtained by collecting feedback on a representative random sample of data items (Equation 7). Equation 1 is used to compute the criteria estimates, and the associated margin of error is calculated with Equations 2, 3 and 4, to obtain the confidence intervals (CIs) for each criterion and for each data source, as shown in Figure 4 (a).

Given these initial estimates, the goal now is finding the combination of sources that maximises the desired criteria (relevance and accuracy) while considering the trade-off between them. This objective can be formulated as a Min-sum model using Equation 13 with the criteria estimates as the coefficients and the number of data items from each source as the decision variables. Min-sum then finds the combination of data items returned by each data source that yields the maximum overall weighted utility oU for the optimisation goal (maximum combined relevance and accuracy).

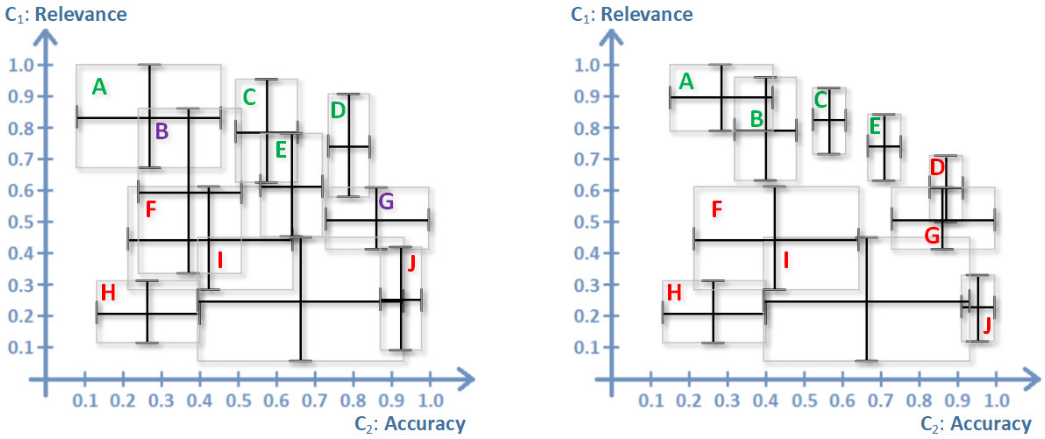
By applying Min-sum over the candidate sources, assume the following subset of data sources contributing to the solution is found: $Ss = \{A, C, D, E\}$. This preliminary solution is illustrated in Figure 4 (a). In the same figure, a different subset of data sources $So = \{B, G\}$ is identified that are not part of Ss but that have CIs for the optimisation criteria that overlap with data sources in Ss (graphically, the areas defined by the CIs of B and G overlap those defined by the CIs of the sources in the non-dominated solution, whereas F, H, I and J do not). This overlap is computed using Equations 6 and 5. It suggests that, in addition to the sources in the preliminary solution Ss, more feedback is needed on B and G in order to decide whether they belong to the solution or not. The proposed TFC strategy then collects more feedback on a new set, $S' = Ss \cup So = \{A, B, C, D, E, G\}$. The data sources in S' benefit from additional feedback either by reducing the uncertainty as to whether or not they should contribute to the solution, or by refining their criteria estimates.

Having decided on which data sources more feedback needs to be collected, the next step is to determine how much feedback should be collected. This is obtained with Equation 7, which computes the sample size over a population that, in this case, consists of the unlabelled data items produced by all the data sources in S'.

Once the number of data items on which feedback needs to be collected has been determined, feedback is collected on them and is used to refine the criteria estimates; this is done by recalculating the estimates and margin of error for each criterion for each data source. This approach is followed for the data sources that are either part of a preliminary solution or are candidates to become part of the solution. This refinement continues while there is enough budget b for feedback collection, there is still overlap between the CIs of data sources contributing to the solution and those from non-contributing sources, and there are unlabelled items.

It is important to notice that in Figure 4 (a), the data sources F, H, I and J are not considered for further feedback collection, since their CIs do not overlap with those of any of the sources contributing to the solution, and therefore, they have no statistical possibility of being part of it, unless there is a need for more data items than those produced by sources in S'. By filtering out these outliers, TFC focuses on those data sources that can be part of the solution.

The strategy leads to a state where the CIs for data sources in the solution do not overlap with the CIs of other data sources, as shown in Figure 4 (b). The result is a solution with a subset of data sources $\{A, B, C, E\}$ with low error estimates, and another subset of data sources $\{D, F, G, H, I, J\}$ that were excluded from additional feedback collection as they have a low likelihood of being part of an improved solution. Notice that, in this example, the data source D, that contributed to the preliminary solution in Figure 4 (a), turned out to be of lower quality once its estimates were refined and was excluded from the improved solution in Figure 4 (b). On the other hand, the data source B, that did not contribute to the preliminary solution in Figure 4 (a), was finally included in the solution in Figure 4 (b) as its estimates turned out to be higher after more feedback was collected.



(a)

(b)

Fig. 4. Confidence intervals with overlapping (a) and without overlapping (b)

In summary, the TFC strategy addresses the problem by:

- Obtaining reliable criteria estimates for a set of candidate data sources based on the feedback provided by users over the values of the data items produced by these candidate data sources.
- Computing a solution to the multi-criteria data source selection problem given the criteria estimates.
- Analysing the overlap between the confidence intervals for the criteria estimates for the candidate data sources, to identify which data sources deserve to be considered for more feedback collection.
- Refining the criteria estimates of the candidate data sources, in the light of the feedback, to improve the solution returned by the optimisation algorithm.

An important feature of the proposed TFC strategy is that it can be applied to problems with multiple criteria, varied user preferences (weights) for each criterion, and over a large number of data sources of variable size and quality.

4.2 Algorithm

This section describes the algorithm for the TFC strategy applied to the MCSS problem. The pseudo-code for the algorithm is given in Figure 5.

The inputs for the algorithm are: S: the collection of data sources from which a subset must be selected that together satisfy the user requirements considering the criteria and specific preferences; C: the collection of criteria modelled as described in Subsection 3.1; U: the set of unlabelled data items produced by data sources in S; W: the collection of criteria weights representing users preferences; b: the allocated budget for the total number of items on which feedback can be obtained; R: the total number of user requested data items; for statistical estimations, cL: the confidence level; and e: the initial margin of error. The output is a vector $\mathbf{X}$ with the number of data items each data source contributes.

To solve the MCSS problem, based on the criteria estimates refined by the proposed TFC approach, the first step is to obtain the sample size for the number of data items that need to be annotated to achieve a statistically representative view of all the data items (line 3). This sample size is computed with Equation 7 using the number of unlabelled data items produced by data sources in S' to represent the sample population. The confidence level and margin of error determine the sample size of the data items on every data source.

```
Input: set of data sources $S$
Input: set of criteria $C$
Input: set of unlabelled data items $U$
Input: set of weights for the criteria $W$
Input: a budget size $b$
Input: a number of user requested data items $R$
Input: a confidence level $cL$
Input: a margin of error $e$
Output: vector with number of data items contributed by each data source $\mathbf{X}$
 1: $L \leftarrow \{\}, \mathbf{X} \leftarrow \{\}, S' \leftarrow S, sS \leftarrow 1$
 2: while $b > 0$ and $sS > 0$ and $S' <> \{\}$ do
 3:     $sS \leftarrow \text{computeSampleSize}(S', U, b, cL, e)$
 4:     $L \leftarrow \text{collectFeedback}(S', U, sS)$
 5:     $b \leftarrow b - sS$
 6:     $\hat{C} \leftarrow \text{estCriteriaValues}(S, L, U, C, cL, e)$
 7:     $\mathbf{X} \leftarrow \text{solveMCSS}(S, \hat{C}, W, R)$
 8:     $S' \leftarrow \{\}$
 9:     for $s \in S$ do
10:         $oL \leftarrow true$
11:         for $c \in C$ do
12:             $oL \leftarrow \text{isSignificantlyOverlapping}(s_c, \mathbf{X})$ and $oL$
13:         end for
14:         if $\mathbf{X}_s > 0$ or $oL$ then
15:             $S' \leftarrow S' \cup s$
16:         end if
17:     end for
18: end while
19: return $\mathbf{X}$
```

Fig. 5. TFC algorithm

The feedback collection process is represented by the *collectFeedback* function (line 4), which takes as arguments the set of sources considered for feedback S', the set of unlabelled data items U, and the number of additional data items on which feedback is required sS. This function randomly selects from U at most sS data items on which feedback needs to be collected. The remaining budget is updated accordingly depending on the number of data items identified (line 5).

The criteria values can be estimated for each candidate data source in S once the feedback is collected and assimilated. The function *estCriteriaValues* (line 6) uses the candidate data sources S, the sets of labelled and unlabelled data items L and U, the collection of criteria C, and a given confidence level and margin of error cL and e, to compute the collection of estimated criteria values $\hat{C}$ for each data source. These estimates rely on the data items already labelled and are computed with Equation 1, as described in Subsection 3.1. The estimates obtained are used to build the confidence intervals (Equations 3 and 4) around each criterion estimate (and for each data source), by computing the margin of error with Equation 2. The confidence intervals are then analysed for overlapping one dimension at a time. This approach is followed to handle multiple dimensions without considering them all at the same time for the statistical computations. An example of how the confidence intervals may look at this early stage of the process is shown in Figure 1(b) from Section 2, where there is high overlapping between the confidence intervals and no clear candidate sources.

At this point, with initial estimated criteria values for all the candidate sources, the MCSS problem can be solved by applying an optimisation model as described in Subsection 3.3 (Min-sum) to obtain a solution that maximises the overall weighted utility oU. The *solveMCSS* function (line 7) represents this step and requires the collection of candidate data sources S, the set of estimated criteria $\hat{C}$, the set of weights representing the user preferences W, and the total number of user requested data items R. The output from this optimisation is a vector $\mathbf{X}$ with the number of data items each data source contributes. The set S' is initialised before processing the data sources (line 8).

Having determined the confidence intervals for each criterion and data source, and the data sources that contribute to a preliminary solution $\mathbf{X}$, the overlap between these intervals can be analysed. This analysis is performed by the *isSignificantlyOverlapping* function (line 12), that takes the estimate for each criterion c in C applied to each data source s in S and the solution for the MCSS problem $\mathbf{X}$ to determine which intervals from data sources contributing to the solution significantly overlap with intervals from non-contributing data sources. The overlapping analysis uses the concepts defined in Subsection 3.2, in particular Equations 5 and 6, to determine if two intervals are significantly overlapping or not. As this overlapping is evaluated at data source level (not criterion level), when all the criteria is evaluated with significant overlap the data source s is considered for feedback collection (condition: *and oL* in line 12).

The next step is for each data source contributing to the solution or for each non-contributing source that has some significant overlap with data sources contributing to the solution (line 14), to be added to the set S' which holds the data sources on which feedback needs to be collected (line 15). S' is used in the next cycle to compute a new sample size sS over the remaining unlabelled data items. After a few rounds of feedback collection the scenario can be as in Figure 4 (a), where there is still some overlapping but the data sources contributing to the solution are now mostly identified.

The iteration continues while any of the following conditions hold (line 2): (i) There is overlapping between confidence intervals of data sources that contribute to the solution and data sources that do not contribute. (ii) The number of data items on which to collect additional feedback obtained by using Equation 7 is greater than zero. In other words, some data items are still left for feedback collection. (iii) The remaining budget b is greater than zero.

When the loop exits, the solution $\mathbf{X}$ (line 19) is a collection of counts of the number of data items to be used from each candidate data source in S. Figure 4 (b) presents a potential image at this stage, where no overlapping exists between confidence intervals of data sources contributing and not contributing to the solution. This condition indicates that the selected data sources should not change if additional feedback is collected. This is, thus, a well-founded approach to deciding when additional feedback is unlikely to be fruitful.

5 EVALUATION: TFC VS RANDOM AND UNCERTAINTY SAMPLING

This section presents the experimental results for evaluating TFC against two competitors: random and uncertainty sampling. Random acts as a baseline. Uncertainty sampling is a general active learning-based technique that is applicable to the explored setting. To the best of our knowledge, there are no specific contributed solutions to this problem in the research literature.

5.1 Competitors

The *random* sampling does not target specific data items for feedback. This baseline competitor considers, as candidates for feedback, all unlabelled items produced by the data sources, providing an even distribution of the feedback collected.

Uncertainty sampling follows the active learning paradigm, which is based on the hypothesis that if a learning algorithm is allowed to choose the information from which it is learning then it will perform better and with less training [Settles 2012]. The decision to incorporate and active learning technique as a competitor of TFC it is based on both strategies seeking to reduce uncertainty, although TFC also considers the user's requirements. In the uncertainty sampling technique, an active learner poses questions to an oracle over the instances for which it is less certain of the correct label [Lewis and Gale 1994]. Often the uncertainty is represented by a probabilistic model that represents the degree of uncertainty associated with the instances. In this paper, the uncertainty is represented by a heuristic that considers the weights of the data criteria and the margins of error for the estimated criterion of a data source. Feedback is collected first on those data items whose originating data source has the largest margin of weighted error, thus taking into account the importance the user places on the criterion. The uncertainty is computed as follows:

$$u_t = \max(w(\hat{c}_{k_s}) \cdot e(s_t, \hat{c}_{k_s})); k = 1, 2, \ldots, n \tag{14}$$

where t is a data item produced by the data source s, u is the uncertainty value on which the items are ranked, w is the data criterion weight, e is the margin of error (Equation (2)), $\hat{c}_{k_s}$ is the data criterion, and n is the number of criteria.

For feedback collection, those items with the highest uncertainty are targeted first, considering all criteria and candidate data sources. For example, to evaluate the uncertainty on two data items (t_1 and t_2) each produced by two data sources (S_1 and S_2, and S_2 and S_3, respectively), and assuming that we have two criteria weighted as follows: $w(C_1) = 0.4$ and $w(C_2) = 0.6$, and that the margins of error for the candidate data sources and criteria are: $e(S_1, C_1) = 0.345$, $e(S_2, C_1) = 0.172$, $e(S_3, C_1) = 0.394$, $e(S_1, C_2) = 0.261$, $e(S_2, C_2) = 0.108$, and $e(S_3, C_2) = 0.289$. By using Equation (14)) the following uncertainty values for each data item are obtained:

$$\begin{aligned} u(t_1) = \max(&w(C_1) \cdot e(S_1, C_1), w(C_1) \cdot e(S_2, C_1), \\ &w(C_2) \cdot e(S_1, C_2), w(C_2) \cdot e(S_2, C_2)) = 0.1566 \\ u(t_2) = \max(&w(C_1) \cdot e(S_2, C_1), w(C_1) \cdot e(S_3, C_1), \\ &w(C_2) \cdot e(S_2, C_2), w(C_2) \cdot e(S_3, C_2)) = 0.1734 \end{aligned}$$

Therefore, t_2 is selected for feedback collection to reduce the higher uncertainty associated with data sources S_2 and S_3.

5.2 Experimental Setup

5.2.1 Data Sources. Taking into account the practical example introduced in Section 1, the following evaluation uses a data set about food products[2]. This data set contains nutritional information about world food products in an open database format. The information has been gradually collected from vendor's websites and added to the database by unpaid contributors. An additional data set (about UK real estate data) was used to evaluate the proposed TFC approach but since the results were very similar to those presented here, these were not included as well.

These experiments take into account 86,864 different data items produced by 117 virtual data sources (where each virtual data source represents a contributor to the database). Each data item has the following textual attributes: creator, product_name, quantity, countries, serving_size and nutrition_score.

The targeting approaches were tested with different numbers of data criteria (two, four and six) and varied weights among them (users preferences). The data criteria considered were (in order): correctness, relevance, usefulness, consistency, conciseness and interpretability. The weights corresponding to each tested scenario and for each data criterion are presented in Table 1.

[2] https://world.openfoodfacts.org/data

Table 1. Criteria weights for experimental scenarios

	2 criteria ($\mathbf{w}_1$)	4 criteria ($\mathbf{w}_2$)	6 criteria ($\mathbf{w}_3$)
Accuracy (c_1)	0.5	0.4	0.3
Relevance (c_2)	0.0	0.2	0.1
Usefulness (c_3)	0.0	0.1	0.2
Consistency (c_4)	0.5	0.3	0.1
Conciseness (c_5)	0.0	0.0	0.2
Interpretability (c_6)	0.0	0.0	0.1

To implement the data criteria several notions were synthetically defined to represent a request or intention. For instance, the notion of relevance is validated against the values of the attribute countries, considering that these data items should be produced in either France or Britain to be considered relevant. Similar notions were defined for the other data criteria in order to build a collection of data sources with varied scores on each dimension, that may represent a real-world data integration scenario.

The experiments were repeated 20 times to reduce the fluctuations due to the random sampling, and the average values were reported. All the statistical computations assume that the data for every data source is normally distributed, and are based on a 95% confidence level ($z - score = 1.96$) with an error of 0.05.

For these experiments, the feedback collected is simulated by sampling over the ground truth, in order to evaluate the performance of the approaches without considering additional factors like worker reliability in crowdsourcing [Liu et al. 2012]. The ground truth was obtained by modelling a typical user's intention over the food data and evaluating this intention across the six data criteria in Table 1.

In these experiments, the maximum overall weighted utility oU is evaluated by applying the Min-sum model from Equation 13 to solve the MCSS problem. This metric is defined in Subsection 5.2.2. A single optimisation technique is used to compare the impact of changing the approach used to refine the estimated values of the criteria by following different feedback collection strategies.

5.2.2 Optimisation Evaluation Metrics. To compare the solutions returned by the optimisation algorithms (described in Section 4) first one needs to define the metric used to evaluate these solutions. This evaluation uses the *overall weighted utility oU* which is a measure of the utility of a solution considering users preferences.

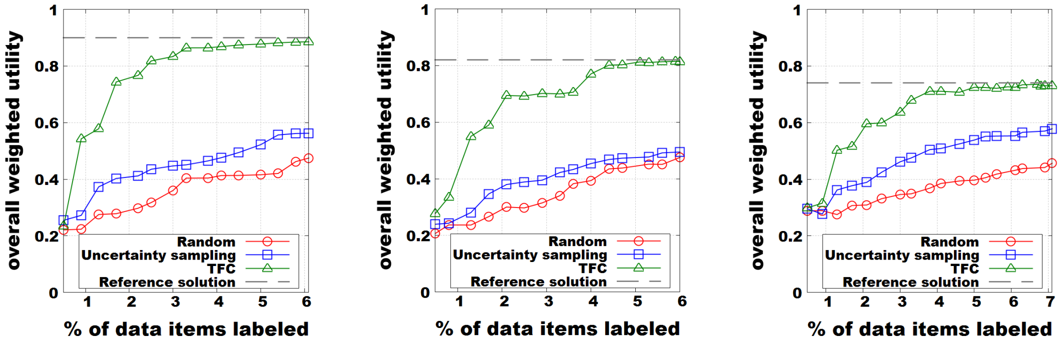
To obtain the oU the average weight-adjusted utility U is computed for each criterion c and considering the range of possible values c can take. The following formula computes U for a given criterion c and a data source s:

$$U_c = \frac{\frac{(\sum_{s=1}^{|S|} x_s \cdot c_s) - Z_c^{**}}{Z_c^{*} - Z_c^{**}}}{R} \tag{15}$$

And then U is combined for all criteria while considering users preferences (weights W) for each criterion c to compute oU:

$$oU = \sum_{c=1}^{n} U_c \cdot W_c \tag{16}$$

where n is the number of data criteria, c_s is the current value of the criterion c for data source s, x_s is the number of data items chosen from data source s, R is the user requested number of data items, Z_c^{*} is the ideal solution value for criterion c, and Z_c^{**} is the negative ideal solution value for c.



(a) (b) (c)

Fig. 6. Results summary for MCSS experiments for (a) 2, (b) 4, and (c) 6 criteria.

5.3 Results

5.3.1 Overall Weighted Utility Comparison. The plots presented in Figure 6 show the *oU* for the three targeting strategies compared on three different scenarios for which the weights are given in Table 1: two criteria with even weights ($\mathbf{w}_1$), four criteria with uneven weights ($\mathbf{w}_2$) and six criteria with varied weights ($\mathbf{w}_3$).

Figure 6 (a) shows the comparison of the averaged *oU* for the three targeting strategies with incremental levels of feedback for two criteria. The dotted line represents a reference solution achieved *without uncertainty* (100% of the data items labelled). The results are clearly favourable for TFC as it finds a solution with more than 0.8 *oU* with only 2.5% of the data items labelled, in comparison with 0.32 and 0.43 *oU* achieved by the random and uncertainty sampling approaches, respectively, for the same amount of feedback collected. As this scenario considers only two criteria ($\mathbf{W}_1$) the solution is hard to find, because the number of potential solutions is larger than when more criteria need to be balanced, in other words, if the number of constraints increases (by having more criteria to select the data items), the number of potential solutions decreases which, in turn, reduces the complexity of the optimisation problem.

In Figure 6 (b), TFC still clearly outperforms its competitors, in a scenario with four data criteria. The averaged overall weighted utility *oU* for the reference solution is not as high as in the previous scenario due to the reduction in the number of potential solutions, which is caused by imposing more restrictions (more criteria) in the optimisation problem. This reduces the difference between the three strategies but, by using TFC, the solution with 2.5% of labelled data items has 0.7 *oU*, while random and uncertainty sampling achieve 0.33 and 0.39 respectively.

Figure 6 (c) shows the averaged *oU* for the scenario with six criteria. In this case, as the number of constraints is increased, the optimisation algorithm finds solutions with lower combined *oU* hence the smaller difference between the three strategies. The advantage of the proposed TFC approach is smaller but, for instance, with 2% of labelled data items, TFC allows a solution with *oU* of 0.6, compared with 0.31 and 0.39 for random and uncertainty sampling respectively.

In the previous figures, the return on investment is clearly favourable for the TFC approach as the overall weighted utility *oU* of the solution achieved by solving the MCSS problem is always larger with TFC, particularly for small amounts of feedback, which is aligned to the objective of reducing the feedback required to obtain effective solutions when following a pay-as-you-go approach.

5.3.2 Feedback Required Comparison. In this section, a comparison between the amount of feedback required for the three approaches to reach the reference solution is presented. The reference solution is considered to be achieved when three successive estimated overall weighted utility values are within a small margin (10^{-6}) of the reference solution.

In Figure 7 (a), the difference between the feedback required to reach the reference solution is compared for the three strategies for two criteria. TFC clearly outperforms the other approaches by a factor of two to four.

In Figure 7 (b), the difference between the feedback required to achieve the results of the reference solution is reduced due to the number of constraints (criteria) considered. With four criteria the number of potential solutions is decreased, therefore, reducing the complexity of the optimisation problem. Even so, the amount of feedback required using the TFC approach is between three and four times smaller than random and uncertainty sampling.

Finally, Figure 7 (c) shows the differences in feedback required over a highly constrained problem (with six criteria), that has a smaller number of potential solutions, compared with the previous two experiments, and therefore the difference between the three strategies is smaller. These results are explained by the fact that the potential solutions for this problem, in the data set used, have a lower oU than the previous experiments and, therefore, the optimisation algorithm easily finds acceptable solutions even when the criteria estimates have a large margin of error. Even with a reduced margin, TFC still outperforms its competitors with low amounts of feedback collected. For instance, the feedback required to reach the reference solution by the TFC strategy is less than half the feedback required by the random approach, and 36% smaller than the feedback required by uncertainty sampling.

A pattern, as we move from Figure 7 (a) to (c), is that uncertainty sampling improves compared with random. As the number of criteria to be balanced increases, the combined weighted utility for the reference solution decreases (i.e., it becomes harder to find very high-utility solutions that correctly balance six criteria instead of only two), hence, by reducing the reference solution threshold, all approaches reach this threshold with less feedback, as the number of combinations of data sources (potential solutions) that can reach the lower threshold increases. In other words, there are more mid-quality than high-quality solutions in the data set.

A characteristic of uncertainty sampling is that it over-focuses the feedback collection on those data sources with the highest uncertainty (in contrast with random, which spreads the feedback evenly among all), therefore, with multiple subsets of data sources as potential solutions, it becomes more likely that uncertainty sampling will refine the estimates of one of these subsets to reach the reference solution than for random to refine the estimates of all the data sources to reach the same goal. That is why, as the number of criteria, and thus the number of potential solutions, are increased, uncertainty sampling eventually outperforms random.

In general, random and uncertainty sampling do not perform well in the tested scenario. For random, the number of candidate data sources and data items causes a large spread of the feedback over all the data sources, including those that have very poor criteria values; this feedback dispersion produces slow improvements throughout the process for this approach. In the case of the uncertainty sampling, its results are slightly better than the random approach when comparing the oU at early stages of the feedback collection process, but the improvement is sometimes even slower than random, as this strategy not only keeps considering all the data sources during the whole process but it also causes the feedback to be wasted on labelling data items from data sources that are not part of the solution, just because those data sources may have a large uncertainty in their criteria estimates.

6 ACCOMMODATING UNRELIABLE FEEDBACK

In this section, an approach is proposed that enables the TFC method from Section 4 to take account of unreliable feedback. In Subsection 6.1, different approaches to unreliable feedback are considered, and in Subsection 6.2 it is shown

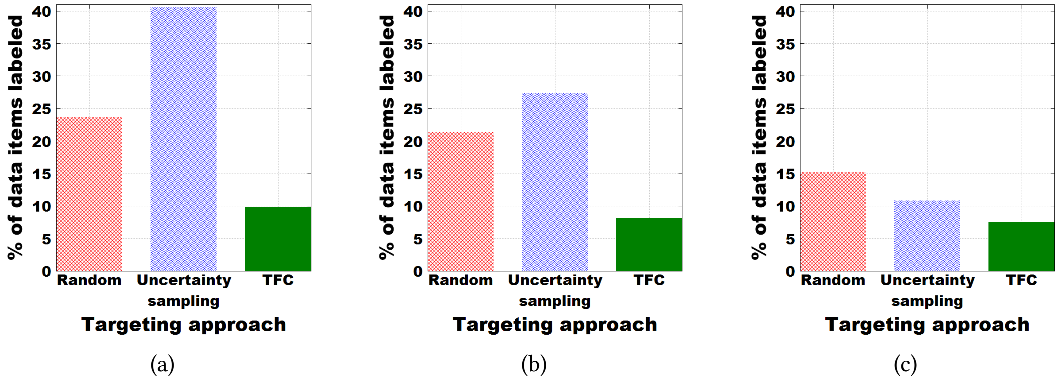


Fig. 7. Results summary for MCSS experiments, comparing the feedback required to reach the reference solution with (a) 2 criteria, (b) 4 criteria, and (c) 6 criteria.

how the statistical foundations of TFC can be extended to support one of these. The proposed TFC strategy is then evaluated with crowd workers in Section 7.

6.1 Approaches to Unreliable Feedback

To extend TFC to accommodate unreliable feedback, the following questions must be answered:

(1) How is the reliability of the crowd workers to be evaluated?
(2) How is the worker's reliability to be assimilated into the TFC approach?

Regarding the first question, in the literature there are already several approaches to dealing with worker reliability, including:

- Majority voting: [Fink 2002; Litwin 1995; Osorno-Gutierrez et al. 2013];
- Probabilistic modelling: [Crescenzi et al. 2013; Zhang et al. 2015];
- Iterative learning: [Karger et al. 2011; Sheng et al. 2008]; and
- Expectation maximisation: [Dawid and Skene 1979; Dempster et al. 1977; Raykar et al. 2010].

Among these options, the simplest in terms of mathematics and also to implement is *majority voting*. To confirm that the proposed TFC strategy can be used with different approaches to evaluate worker reliability, results are compared using majority voting and expectation maximisation in Section 7. Other alternatives are based on assumptions that do not apply to the problem of evaluating the worker's reliability or need initialisation and/or training.

For the second question, there are also some alternatives on how to assimilate the obtained reliability:

- Apply an *individual* worker's *reliability* at *data item-level* (this option is computationally-intensive, and item-level feedback is not especially well matched for collection-based source selection).
- Apply *individual* worker's *reliability* at *data source-level* (requires data items to be pre-selected to build surveys).
- Apply *overall reliability* at *data source-level* (eases reliability assimilation, no pre-selection needed).

As decisions are made in TFC based on source level estimates for criteria values, the *overall reliability at data source-level* is adopted, which also turns out to be straightforward to implement in practice.

6.2 Incorporating Unreliable Feedback

As in previous works [Fink 2002; Litwin 1995; Osorno-Gutierrez et al. 2013], there is the need to generate a subset of redundant data items that will be assigned to different workers (*Inter Observer Reliability* or IrOR). Majority voting is then used to assess the workers' reliability. To evaluate each worker's reliability for a given data source S the labels for IrOR are compared, counting how many times each worker agrees or disagrees with the other workers, and computing the reliability as follows:

$$worker\_reliability(S) = \frac{\#agreed(S)}{\#agreed(S) + \#disagreed(S)} \tag{17}$$

$$overall\_reliability(S) = \frac{\sum_{i=1}^{\#workers} worker_i\_reliability(S)}{\#workers} \tag{18}$$

All workers with a reliability below rT are excluded and their responses discarded; this deals with systematic errors from workers trying to complete tasks quickly, or incorrectly on purpose.

In statistical theory, there is the standard error of measurement sem [Bulmer 1979] that estimates the extent to which an instrument (e.g. a task) provides accurate responses. It requires the standard deviation σ of the instrument and its reliability r:

$$sem = \sigma \cdot \sqrt{1 - r} \tag{19}$$

For a population proportion p, σ can be replaced by the standard error for a sample size n [Bulmer 1979]. Equation (19) can be rewritten as:

$$sem = \sqrt{\frac{p \cdot (1 - p)}{n}} \cdot \sqrt{1 - r} \tag{20}$$

For the proposed TFC strategy, to calculate the confidence intervals with sem instead of the standard error se Equation (2) is modified as follows:

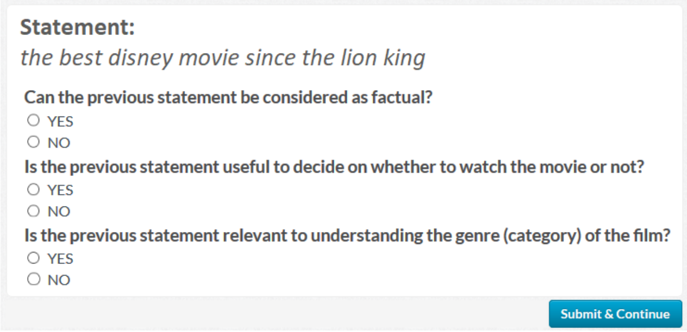
$$\begin{aligned} e*(cL) &= z_{cL} \cdot sem(S, L, \hat{c}, r) \cdot fpcf(S, T, L) \qquad (21) \\ &= z_{cL} \cdot \left(\sqrt{\frac{\hat{c}_S \cdot (1 - \hat{c}_S)}{|L_S|}} \cdot \sqrt{1 - r_S} \right) \cdot \left(\sqrt{\frac{|T_S| - |L_S|}{|T_S| - 1}} \right) \end{aligned}$$

where: $\hat{c}$ is the estimated user criterion (proportion), S is a single data source or a set of sources, r is the reliability of the feedback collected, $e*$ is the adjusted margin of error (considering reliability), cL is the confidence level required, L is the set of distinct data items labelled (feedback), T is the total number of data items produced by the data sources, and z is the z–score for a confidence level (assuming a normal data distribution).

7 EVALUATION: TFC WITH UNRELIABLE WORKERS

This section describes an experimental evaluation of the approach from Section 6 using crowdsourced micro-tasks hosted by CrowdFlower (now Figure Eight)[3].

[3] https://www.figure-eight.com/



Statement:

the best disney movie since the lion king

Can the previous statement be considered as factual?

○ YES

○ NO

Is the previous statement useful to decide on whether to watch the movie or not?

○ YES

○ NO

Is the previous statement relevant to understanding the genre (category) of the film?

○ YES

○ NO

Fig. 8. An example form showing how crowd workers submit feedback

7.1 Crowdsourcing Task

The task used in these experiments involves consulting crowd workers to identify data sources of movie reviews, on the basis of whether they are factual, useful and relevant. A statement is considered to be: *factual* if it conveys no opinion, and simply provides information; *useful* if it provides information that can help in making a decision as to whether or not to watch a movie; and *relevant* if it provides information on the type or genre of a movie. The overall objective is thus to identify sources that contain suitable movie reviews.

For this task, an existing data set is used, that included movie reviews from different data sources[4], and designed tasks that asked users about the factuality, usefulness and relevance of statements from the reviews[5]. The crowd worker is provided with detailed instructions on what is required, along with examples, and is then asked questions about statements from reviews, using the web form illustrated in Figure 8.

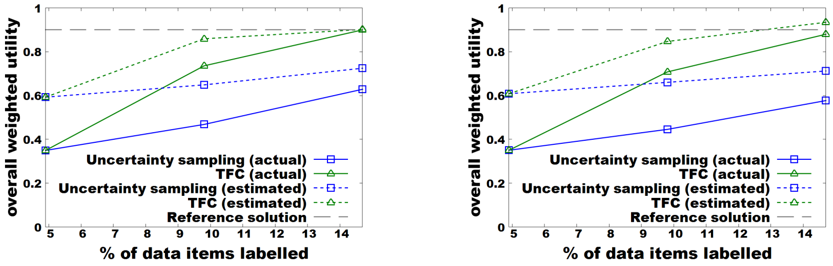
7.2 Experiment Setup

The experiment involved the use of *67* data sources containing 10,000 distinct reviews. The number of iterations for feedback collection was three – one involving random sampling for data source annotation and worker reliability assessment, and then two with targeted collection of additional feedback. In the initial phase, any worker with a reliability less than *0.7* is rejected as untrustworthy.

Each task obtains feedback on 30 statements from reviews, of which 27 are asked to only one worker and three are shared with others for estimating worker reliability. The time that a task would take to complete is estimated, and workers were paid at the rate of the UK National Minimum Wage (which resulted in a payment of £3.72 per 30-question task). In this scenario, the proposed TFC approach is compared against the active learning-based technique called Uncertainty sampling, described in Subsection 5.1 to demonstrate how an approach that focuses on reducing the uncertainty does not necessarily provide improved solutions in a scenario where the uncertainty is not only in the criteria estimates but also in the feedback used to refine those estimates. To provide a comparison between the proposed TFC approach with and without considering the unreliability of the collected feedback, an additional experiment was carried out involving synthetic feedback, i.e., provided by gradually disclosing the ground truth, but using the same data set about movie reviews and an equivalent experimental setup as the crowdsourced experiment.

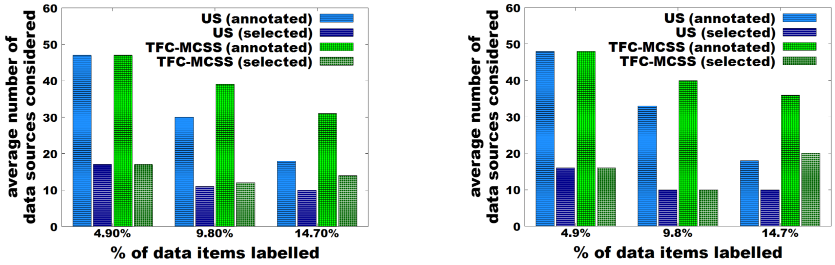
[4] http://www.cs.cornell.edu/People/pabo/movie-review-data/

[5] In this data set, batch numbers are related to how and when the reviews were collected, and consider data with different batch numbers to represent different data sources. The resulting data sources are varied in terms of quality and size.



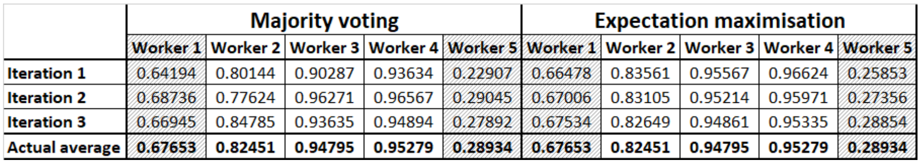
(a) (b)

Fig. 9. TFC and Uncertainty Sampling for MCSS using (a) synthetic and (b) crowdsourced feedback



(a) (b)

Fig. 10. Sources considered using (a) synthetic and (b) crowdsourced feedback



	Majority voting					Expectation maximisation				
	Worker 1	Worker 2	Worker 3	Worker 4	Worker 5	Worker 1	Worker 2	Worker 3	Worker 4	Worker 5
Iteration 1	0.64194	0.80144	0.90287	0.93634	0.22907	0.66478	0.83561	0.95567	0.96624	0.25853
Iteration 2	0.68736	0.77624	0.96271	0.96567	0.29045	0.67006	0.83105	0.95214	0.95971	0.27356
Iteration 3	0.66945	0.84785	0.93635	0.94894	0.27892	0.67534	0.82649	0.94861	0.95335	0.28854
Actual average	**0.67653**	**0.82451**	**0.94795**	**0.95279**	**0.28934**	**0.67653**	**0.82451**	**0.94795**	**0.95279**	**0.28934**

Fig. 11. Overall crowd workers' reliability

To evaluate the performance of majority voting to estimate the workers' reliability, an experiment was carried out to compare the reliability obtained also through *expectation maximisation* [Dawid and Skene 1979], which is a popular algorithm to estimate the quality of an element in the presence of incomplete evidence. The results for a subset of crowd workers providing the feedback for the three iterations from the previous experiment are shown in Figure 11. In this table, the workers marked with diagonals were discarded due to having a reliability below the threshold (rT) of 0.7. As can be observed, the computed reliability is close enough to discard the same workers in both scenarios. Although, the expectation maximisation algorithm relies on further iterations to converge towards the actual workers' reliability, this technique benefits from several rounds of feedback collection to further refine their estimates. Therefore, a decision was made to maintain the majority voting technique to estimate the workers' reliability since its simplicity produces acceptable results for the purposes of the overall TFC approach while requiring a reduced number of feedback collection rounds.

7.3 Results

Figure 9 presents the actual and estimated overall weighted utility for uncertainty sampling and TFC, where feedback is collected synthetically (a), and through the crowdsourcing platform (b). The following can be observed:

- The initial overall weighted utility *oU* is the same for both approaches, as they both share the same initial sample of responses from crowd workers that provide preliminary estimates of the criteria for each source.
- When additional feedback is collected, the results obtained by TFC are significantly better than those from uncertainty sampling. This is at least in part because TFC considers fewer sources to be candidates for inclusion than uncertainty sampling, as illustrated in Figure 10.
- As more feedback is assimilated, the difference between the actual and estimated *oU* drops faster for TFC than for uncertainty sampling in all scenarios. This is because of a combination of two factors: (i) the feedback is targeted on the included data sources using TFC; and (ii) the amount of feedback collected on each data source takes into account the reliability of the workers providing the feedback.
- Majority voting, used to estimate the workers' reliability, provides acceptable results (compared with expectation maximisation), as it can be used to discard the feedback provided by unreliable workers, as shown in Figure 11.
- Where reliable feedback is obtained synthetically (Figure 9 (a) and Figure 10 (a)), the results are consistent with those obtained in the previous experiments with reliable feedback in Subsection 5.3. Through the TFC strategy, the reference solution (solution without uncertainty) is reached with 15.9% of the data items labelled, when the unreliable feedback is handled by the mechanisms described in Subsection 6.2. If synthetic feedback is used, TFC helps reaching the reference solution with 14.3% of labelled items. The difference derives from the additional 10% of feedback required to measure and handle the workers' reliability. In the experiment with synthetic feedback, the error reduction is significantly faster, as all the feedback is used to refine the estimates.

8 RELATED WORK

This section discusses the relevant related work for the problem described in Section 2, specifically on multi-criteria data source selection and how to identify the data items on which feedback should be collected to improve the results.

8.1 Data source Selection with Multiple Criteria

Most work on data source selection uses one or two dimensions. In [Dong et al. 2012], the source selection is guided by the marginal gain in the financial value of the selected sources, with the predicted economic value of a source based on the estimated accuracy of its contents. Thus, the optimisation problem is one dimensional (marginal gain), and other criteria are combined in a way that reflects their contribution to the value of the data. In [Rekatsinas et al. 2015], the objective is to automatically infer the sources quality following a vision which was later implemented as the SourceSight system [Rekatsinas et al. 2016]. This research is the most related to the MCSS technique that forms the context for our work, even though the result from SourceSight is not a single solution but a set of non-dominated solutions with different trade-offs from which the user can select one. SourceSight does not investigate feedback targeting. The method proposed in [Talukdar et al. 2010] follows the pay-as-you-go approach to consider new and existing feedback on search results to construct queries over new and existing sources, in a setting where active learning has been used for feedback targeting [Yan et al. 2015]. Concerning solutions involving multiple dimensions, few works, as far as we know, have addressed the problem. In [Rekatsinas et al. 2014], the objective is to select a subset of the candidate data sources with the highest perceived gain. This gain is inferred by considering a fixed number of criteria such as accuracy, freshness and coverage to evaluate and rank the different sources; user preferences are not considered in this work. Another example is the approach followed in [Mihaila et al. 2000, 2001], where several criteria are considered for the source selection problem, such as completeness, recency, frequency of updates and granularity. These criteria are evaluated for ranking purposes using a data model that represents the quality of the data in the form of meta-data associated with the candidate sources. This work assumes that there is no uncertainty about the values of those quality dimensions and the fuzziness is used to match the user preferences to the quality meta-data. In [Martin et al. 2014], the quality criteria considered are completeness, consistency, accuracy, minimality, and performance, applied to data integration tasks such as schema matching and mapping refinement, but in this case the criteria are assumed to be certain.

The MCSS proposal for which we target feedback [Abel et al. 2018] is outlined in Subsection 3.3. The paper discusses several optimisation strategies, and the trade-offs they make. This paper extends the work of [Abel et al. 2018] by describing an approach to feedback targeting for such weighted multi-dimensional optimisation problems.

8.2 Targeted Feedback Collection

This section reviews different approaches to targeted feedback collection. The discussion is in two main categories, the ones that follow generic strategies like active learning, and those relying on bespoke techniques.

The *active learning* approaches consider that a machine learning algorithm can perform better if it is allowed to identify its own training data [Settles 2012], by determining which questions to present to an oracle who provides the training data (e.g. a crowd worker). *Uncertainty sampling* is a technique that follows the active learning paradigm, which targets those elements with the highest associated uncertainty. This uncertainty can be represented in different ways depending on the problem at hand. In [Lewis and Gale 1994], uncertainty sampling is used for data classification, and the uncertainty is computed based on the probability that a data pattern belongs to a certain class. In a similar way, uncertainty sampling can indicate which elements need to be considered for feedback collection and, consequently, reduce the uncertainty. For this reason, this technique was selected as a competitor to the TFC strategy.

Among other works that have applied active learning in data management problems, there is a solution, designed for entity resolution, called Corleone [Gokhale et al. 2014], where the uncertainty is used to select pairs of records to collect feedback about for which decision tree classifiers exhibit more disagreement. This work (like TFC) has also provided ways to decide at which point enough feedback has been collected. In the context of web data extraction, [Crescenzi et al. 2015] presents a solution that collects feedback on the results produced by automatically generated extraction rules, using a strategy based on vote entropy [Settles 2012] to evaluate the probability of correctness of these rules. For classification tasks of larger magnitude, the work in [Mozafari et al. 2014] focuses on ways to apply active learning techniques for crowdsourcing, addressing problems like obtaining collections of results back from the crowd, dealing with unreliable workers, and producing a generalised approach to tackle arbitrary classification scenarios.

Some approaches require particular and strong assumptions, closely related to each problem tackled. For instance, in synthesising record linkage rules [Isele and Bizer 2012], the assumption is that a single element from the set of candidates is required (the correct or the most suitable one) and the others can, therefore, be discarded. In our case, multiple data sources may be required, and data sources are removed from consideration only when enough evidence has been gathered to conclude that they should not contribute to the solution.

Despite the large number of fields and applications where active learning techniques have been implemented, there are several data management problems where researchers have developed tailored solutions to obtain feedback in a selective way. Among these works, there are some crowd database systems like Quirk [Marcus et al. 2011] and CrowdDB [Franklin et al. 2011]; in these systems, a query specifies the result required and the query optimiser determines the minimum number of questions that need to be posed to the crowd, to reduce the associated costs, based on the features of the query and other factors like the reliability of the crowd worker. Regarding skyline queries, in [Lofi et al. 2013] the authors use the crowd to enquire about missing values, based on how the risk of having missing data entries affects the quality of query results. In [Liu et al. 2012], a bespoke solution called Crowdsourcing Data Analytics System (CDAS)

[Liu et al. 2012] uses a pre-established accuracy to predict the number of crowd tasks required to achieve a result, considering also the worker's reliability. As more evidence is collected, the system adjusts the number of required crowd tasks depending on how far away the objective is. In [Goldberg et al. 2017], the proposed pi-CASTLE system is developed to use crowdsourcing in prediction problems involving conditional random fields, applied to text labelling and extraction, using constrained inference to assimilate the feedback provided by the crowd. Another work, that also applies crowdsourcing to entity resolution, is [Chai et al. 2018], in which a novel framework CrowdEC is presented that relies on an incentive-based strategy to evaluate and improve the quality of the crowd workers responses. To analyse how complex is to handle unreliable responses provided by crowd workers, [Zheng et al. 2017] presents a survey on different approaches on real datasets, that try to infer the truth from responses collected in crowdsourced platforms, finding that none of the evaluated algorithms perform with the required accuracy and stability, leaving this problem open for future research. In [Zheng et al. 2015], the Quality-Aware Task Assignment System for Crowdsourcing Applications (QASCA) is introduced, a solution to assign tasks to crowd workers in an efficient manner and using data evaluation metrics such as accuracy and f-score. These are examples of crowdsourcing applied to data integration. In this paper, crowdsourcing was used to obtain the feedback required to reduce uncertainty in criteria estimates.

In our previous work on feedback targeting for source selection [Ríos et al. 2016], feedback was identified to support source selection in which the problem to be solved was to maximise precision for a given level of recall (or *vice versa*). In this approach, the sources were sorted by their estimated precision, and feedback was collected on all sources that were candidates to contribute to the solution. A source was a candidate if its estimated precision (taking into account the margin of error for the estimate) could be above the threshold for inclusion. Here, as in [Ríos et al. 2016], the margin for error is considered in criteria values, but we have extended the approach to the more general setting of weighted multi-criteria data source selection, substantially changing the algorithm for feedback targeting.

Solutions using active learning techniques and other tailored approaches are guided by particular features of the problem. In general, the active learning-based techniques do not seem suitable for our problem because they aim to provide uncertainty reduction over individual elements (data sources), and in our case the optimisation is performed by testing different combinations of data sources; in other words, individual source uncertainty does not help the selection process to find the best combination to satisfy the requirements. That is why a bespoke solution was developed.

9 CONCLUSIONS

This paper presented a TFC strategy for targeting data items for feedback, to enable cost-effective MCSS. TFC addresses the problem of incomplete evidence about the criteria that inform source selection and its key features are:

(1) Feedback is collected in support of multi-criteria optimisation, in a way that takes into account the impact of the uncertainty on the result of the optimisation.
(2) Feedback is collected not for individual data sources in isolation, but rather taking into account the fact that the result is a set of data sources.
(3) Feedback is collected on diverse types of criteria, of which there may be an arbitrary number, and user preferences in the form of weights are taken into account during the targeting.
(4) Feedback collection stops when the collection of further feedback is expected not to change which data sources contribute to a solution (i.e., there is no significant overlap between the criteria estimates for the selected and rejected data sources).

(5) Unreliable feedback is addressed through an extension to the statistical approach applied for accommodating uncertain criterion values in the context of reliable feedback.

(6) Experimental results, with real world data, show substantial improvements in the cost-effectiveness of feedback, compared with a baseline (random) solution and an active learning technique.

ACKNOWLEDGMENTS

Julio César Cortés Ríos is supported by the Mexican National Council for Science and Technology (CONACyT). Data integration research at Manchester is supported by the UK EPSRC, through the VADA Programme Grant.

REFERENCES

Edward Abel, Keane John, Norman W. Paton, Fernandes Alvaro A.A., Martin Koehler, Nikolaos Konstantinou, Nurzety Bintiahmadazuan, and Suzanne M. Embury. 2018. User Driven Multi-Criteria Source Selection. *Information Sciences* 430–431 (2018), 179–199.

Tudor Barbu, Mihaela Costin, and Adrian Ciobanu. 2013. Color-Based Image Retrieval Approaches Using a Relevance Feedback Scheme. In *New Concepts and Applications in Soft Computing*. Springer-Verlag Berlin Heidelberg, Frankfurt, Germany, 47–55. https://doi.org/10.1007/978-3-642-28959-0_3

Khalid Belhajjame, Norman W. Paton, Suzanne M. Embury, Alvaro A. A. Fernandes, and Cornelia Hedeler. 2013. Incrementally improving dataspaces based on user feedback. *Inf. Syst.* 38, 5 (2013), 656–687. https://doi.org/10.1016/j.is.2013.01.006

Alessandro Bozzon, Marco Brambilla, and Stefano Ceri. 2012. Answering Search Queries with CrowdSearcher. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 1009–1018. https://doi.org/10.1145/2187836.2187971

Michael G. Bulmer. 1979. *Principles of Statistics*. Dover Publications, New York, NY, USA.

Chengliang Chai, Ju Fan, and Guoliang Li. 2018. Incentive-Based Entity Collection using Crowdsourcing. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018 (ICDE 2018)*. CPS, California, USA, 12.

Valter Crescenzi, Alvaro A.A. Fernandes, Paolo Merialdo, and Norman W. Paton. 2017. Crowdsourcing for Data Management: a Survey. *Knowledge and Information Systems* 53, 1 (2017), 1–41. https://doi.org/10.1007/s10115-017-1057-x

Valter Crescenzi, Paolo Merialdo, and Disheng Qiu. 2013. Wrapper Generation Supervised by a Noisy Crowd. *DBCrowd 2013* 1, 1 (2013), 8–13.

Valter Crescenzi, Paolo Merialdo, and Disheng Qiu. 2015. Crowdsourcing large scale wrapper inference. *Distributed and Parallel Databases* 33, 1 (2015), 95–122. https://doi.org/10.1007/s10619-014-7163-9

Alexander P. Dawid and Allan M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics 1979* 28(1) (1979), 20–28.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39(1) (1977), 1–38.

Xin Luna Dong, Barna Saha, and Divesh Srivastava. 2012. Less is More: Selecting Sources Wisely for Integration. *PVLDB* 6, 2 (2012), 37–48.

Arlene Fink. 2002. *The Survey Handbook*. SAGE Publications, Inc., University of California at Los Angeles, USA, The Langley Research Institute.

Donald H. Foley. 1972. Considerations of sample and feature size. *IEEE Transactions on Information Theory* 18, 5 (1972), 618–626.

Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. 2011. CrowdDB: Answering Queries with Crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (SIGMOD '11)*. ACM, New York, NY, USA, 61–72.

Abir Gallas, Walid Barhoumi, and Ezzeddine Zagrouba. 2014. Negative Relevance Feedback for Improving Retrieval in Large-Scale Image Collections. In *2014 IEEE International Symposium on Multimedia, ISM 2014, Taichung, Taiwan, December 10-12, 2014*. IEEE, ISM, Taichung, Taiwan, 1–8.

Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F. Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. 2014. Corleone: Hands-off Crowdsourcing for Entity Matching. In *Proceedings of the Int. Conference on Management of Data (SIGMOD '14)*. ACM, New York, NY, USA, 601–612.

Sean Goldberg, Daisy Zhe Wang, and Christan Grant. 2017. A Probabilistically Integrated System for Crowd-Assisted Text Labeling and Extraction. *J. Data and Information Quality* 8, 2, Article 10 (Feb. 2017), 23 pages. https://doi.org/10.1145/3012003

Giovanni Grano, Adelina Ciurumelea, Sebastiano Panichella, Fabio Palomba, and Harald C. Gall. 2018. Exploring the integration of user feedback in automated testing of Android applications. In *25th Int. Conference on Software Analysis, Evolution and Reengineering*. SANER, Campobasso, Italy, 72–83.

Alon Halevy, Flip Korn, Natalya F. Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods: Organizing Google's Datasets. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16)*. ACM, New York, NY, USA, 795–806.

Robert Isele and Christian Bizer. 2012. Learning Expressive Linkage Rules using Genetic Programming. *PVLDB* 5, 11 (2012), 1638–1649.

David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative Learning for Reliable Crowdsourcing Systems. *Advances in Neural Information Processing Systems* 24 (2011), 1953–1961.

Andrea Knezevic. 2008. Overlapping confidence intervals and statistical significance. *Cornell University, Statistical Consulting Unit* 73 (2008), 1–1.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can Neural Machine Translation be Improved with User Feedback?. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, USA*. NAACL HLT, New Orleans, Louisiana, USA, 92–105.

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*. Springer-Verlag New York, Inc., New York, NY, USA, 3–12.

James R. Lewis and Jeff Sauro. 2006. When 100% really isn't 100%: Improving the accuracy of small–sample estimates of completion rates. *JUS* 3(1) (2006), 136–150.

Guoliang Li, Jianan Wang, Yudian Zheng, and Michael J. Franklin. 2016. Crowdsourced Data Management: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (2016), 2296–2319. https://doi.org/10.1109/TKDE.2016.2535242

Mark S. Litwin. 1995. *How to Measure Survey Reliability and Validity.* SAGE Publications, Inc, UCLA School of Medicine, Los Angeles, USA.

Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. 2012. CDAS: A Crowdsourcing Data Analytics System. *PVLDB* 5, 10 (2012), 1040–1051.

Christoph Lofi, Kinda El Maarry, and Wolf-Tilo Balke. 2013. Skyline Queries in Crowd-enabled Databases. In *Proceedings of the 16th International Conference on Extending Database Technology (EDBT '13)*. ACM, New York, NY, USA, 465–476. https://doi.org/10.1145/2452376.2452431

Matteo Magnani and Danilo Montesi. 2010. A Survey on Uncertainty Management in Data Integration. *JDIQ* 2, 1, Article 5 (July 2010), 33 pages.

Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, and Robert C. Miller. 2011. Human-powered Sorts and Joins. *PVLDB* 5, 1 (2011), 13–24.

Nigel Martin, Alexandra Poulovassilis, and Jianing Wang. 2014. A Methodology and Architecture Embedding Quality Assessment in Data Integration. *J. Data and Information Quality* 4, 4, Article 17 (May 2014), 40 pages. https://doi.org/10.1145/2567663

George A. Mihaila, Louiqa Raschid, and Maria-Esther Vidal. 2000. Using Quality of Data Metadata for Source Selection and Ranking. In *Proceedings of the 3rd International Workshop on the Web and Databases, Dallas, Texas, USA, in conjunction with PODS/SIGMOD 2000.* ACM, New York, NY, USA, 93–98.

George A. Mihaila, Louiqa Raschid, and Maria-Esther Vidal. 2001. Source Selection and Ranking in the WebSemantics Architecture: Using Quality of Data Metadata. *Advances in Computers* 55 (2001), 87–118. https://doi.org/10.1016/S0065-2458(01)80027-9

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. 2014. Scaling Up Crowd-Sourcing to Very Large Datasets: A Case for Active Learning. *PVLDB* 8, 2 (2014), 125–136.

M. A. Viraj J. Muthugala and A. G. Buddhika P. Jayasekara. 2017. Enhancing User Satisfaction by Adapting Robot's Perception of Uncertain Information Based on Environment and User Feedback. *IEEE Access* 5 (2017), 26435–26447. https://doi.org/10.1109/ACCESS.2017.2777823

Fernando Osorno-Gutierrez, Norman W. Paton, and Alvaro A. A. Fernandes. 2013. Crowdsourcing Feedback for Pay–As–You–Go Data Integration. In *DBCrowd*. CEUR-WS.org, Riva del Garda, Trento, Italy, 32–37.

Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. 2002. Data Quality Assessment. *CACM-Supp. community & building social capital* 45, 4 (2002), 211–218.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning From Crowds. *Journal of Machine Learning Research* 11 (2010), 1297–1322.

Theodoros Rekatsinas, Amol Deshpande, Xin Luna Dong, Lise Getoor, and Divesh Srivastava. 2016. SourceSight: Enabling Effective Source Selection. In *SIGMOD Conference, San Francisco, CA, USA.* ACM, New York, NY, USA, 2157–2160. https://doi.org/10.1145/2882903.2899403

Theodoros Rekatsinas, Xin Luna Dong, Lise Getoor, and Divesh Srivastava. 2015. Finding Quality in Quantity: The Challenge of Discovering Valuable Sources for Integration. In *CIDR*. CIDR, Asilomar, California, USA, 1–7.

Theodoros Rekatsinas, Xin Luna Dong, and Divesh Srivastava. 2014. Characterizing and selecting fresh data sources. In *SIGMOD*. ACM, New York, NY, USA, 919–930. https://doi.org/10.1145/2588555.2610504

Julio César Cortés Ríos, Norman W. Paton, Alvaro A. A. Fernandes, Edward Abel, and John A. Keane. 2017. Targeted Feedback Collection applied to Multi-Criteria Source Selection. In *New Trends in Databases and Information Systems, ADBIS 2017, Nicosia*. Springer, Cham, Nicosia, Cyprus, 136–150.

Julio César Cortés Ríos, Norman W. Paton, Alvaro A. A. Fernandes, and Khalid Belhajjame. 2016. Efficient Feedback Collection for Pay-as-you-go Source Selection. In *SSDBM*. ACM, New York, NY, USA, 1:1–1:12. https://doi.org/10.1145/2949689.2949690

Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (2012), 1–114.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining (KDD '08).* ACM, New York, NY, USA, 614–622.

Partha Pratim Talukdar, Zachary G. Ives, and Fernando C. N. Pereira. 2010. Automatically incorporating new sources in keyword search-based data integration. In *ACM SIGMOD, USA, 2010.* ACM, New York, NY, USA, 387–398. https://doi.org/10.1145/1807167.1807211

Mauricio Villegas, Luis A. Leiva, and Roberto Paredes. 2013. Interactive Image Retrieval Based on Relevance Feedback. In *Multimodal Interaction in Image and Video Applications*. Springer-Verlag Berlin Heidelberg, Barcelona, Spain, 83–109. https://doi.org/10.1007/978-3-642-35932-3_6

Zhepeng Yan, Nan Zheng, Zachary G. Ives, Partha Pratim Talukdar, and Cong Yu. 2015. Active learning in keyword search-based data integration. *VLDB J.* 24, 5 (2015), 611–631.

Chen Jason Zhang, Lei Chen, Yongxin Tong, and Zheng Liu. 2015. Cleaning uncertain data with a noisy crowd. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015.* CPS, California, USA, 6–17. https://doi.org/10.1109/ICDE.2015.7113268

Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth Inference in Crowdsourcing: Is the Problem Solved? *PVLDB* 10, 5 (2017), 541–552. https://doi.org/10.14778/3055540.3055547

Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. 2015. QASCA: A Quality-Aware Task Assignment System for Crowdsourcing Applications. In *Proceedings of the 2015 SIGMOD International Conference on Management of Data, Melbourne.* ACM, New York, NY, USA, 1031–1046.