Author's response to reviews

Title: Some methods for blindfolded record linkage

Authors:
Dr Tim Churches (tchur@doh.health.nsw.gov.au)
Dr Peter Christen (peter.christen@anu.edu.au)

Version: 3 Date: 8 Jun 2004

We thank all the reviewers for their thorough, knowledgeable and helpful comments. Details of the changes made to the manuscript follow.

Reviewer 1 (Ross Lazarus):
No revisions requested.

Reviewer 2 (Leicester Gill):
Discretionary revisions:
- page 4 para 5: 'anonymous record linkage' - we elected to retain the term 'blindfolded record linkage', preferring to avoid the term 'anonymous' altogether due to the multiplicity of ways in which it has been used in the past.
- page 5 para 5: '..factoring the product...': now corrected.
- page 6 para 4: added a comment regarding big-endian/little endian issues - most hash functions do account for this.
- page 7 para 7: inserted a short note regarding chaffing (and winnowing) as described by Rivest.
- page 10 para 1: added a further note and reference to "chaffing and winnowing", and added a sentence explaining record identifiers better.
- page 12 para 1: no information about 'byte boundaries' of bigrams is required or retained: bigrams are formed merely by advancing a 2 character window across the target string. This is now better illustrated in the worked examples in tables 1 and 2.
- ref 34: corrected.

Reviewer 3(Catherine Quantin):
Major compulsory revisions:
- page 10: explanation about power sets: we have added a paragraph in the section titled "n-gram similarity comparisons" (pages 8 and 9) which explains why subsets are needed for comparison of different strings. Also, Tables 1 and 2 now show a worked example.
- pages 10-11: feasibility of the proposed approach: The paragraph describing the improvements in efficiency which can be gained has been expanded. These improvements do not result in the need
for changes to the bigram score calculation, as the longer bigram sublists (the ones resulting in larger bigram scores) are still being used - rather the gains are due to pruning of short bigram sublists which cannot possibly result in a bigram score above a certain threshold.

- point 4 (page 19): mention bigram score used for matching weight: we have added an explanation that the "similarity score" is, in fact, just the bigram_score as described earlier. No modification is needed when the efficiency mechanism (using a threshold as described on page 12) is applied, because we are merely omitting the comparison with short bigrams, and this omission will not change the values of the maximum bigram score calculated for a given pair of records.

- table 1 (now table 3): a longer caption has been added explaining the contents of the table in more detail.

- appendix 1: Python source code has been included in the appendix, together with comments in the source code explaining what is being done in each part of each programme. Python is quite close to "pseudocode", and we felt that for most readers little was to be gained in terms of brevity or clarity by translating the Python code into some other arbitrary pseudocode.

- appendix 2: the infeasibility of this protocol is already described: "In step 6, a combinatorial explosion results in huge numbers of tuples - for example, several million from only ten original values. Unfortunately this renders the protocol impractical". We have also added a comment that the protocol is impractical in the first part of the appendix. However, we still believe it is useful to include the protocol as a challenge to other researchers to come up with a practical method for achieving the same goal.

Minor essential revisions:
- description of protocols: we elected to leave the descriptions in the body of the paper as they were, but have added a worked example in tables 1 and 2.