|  |
| --- |
| [VLDB 2014](http://vldb.org/pvldb" \t "_blank) **Proceedings of the Very Large Database Endowment, Volume 7** Hangzhou, China |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Meta-Reviews For Paper**   |  |  | | --- | --- | | **Track** | Research -> March 2014 | | **Paper ID** | 907 | | **Title** | Computing Relational Sufficient Statistics for Large Databases |  |  |  | | --- | --- | | **Masked Meta-Reviewer ID:** | Meta\_Reviewer\_1 | | **Meta-Reviews:** |  |  |  |  | | --- | --- | | **Question** |  | | Overall Rating | Reject | | Detailed Comments | This paper presents a way to compute sufficient statistics for column combinations in large relational databases. There are concerns on the scalability of the approach and the limitation of the evaluation against only Bayes Nets. | | A crucial assumption needs better justification | No | | The problem needs better motivation, e.g. through a detailed realistic scenario | No | | Notation has to be cleaned up and/or definitions must be tightened | No | | Presentation must be improved | Yes | | Better comparison is required against currently known methods | Yes | | Experiments must be run against more realistic workloads | Yes | |  |

|  |
| --- |
| [VLDB 2014](http://vldb.org/pvldb) **Proceedings of the Very Large Database Endowment, Volume 7** Hangzhou, China |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Reviews For Paper**   |  |  | | --- | --- | | **Track** | Research -> March 2014 | | **Paper ID** | 907 | | **Title** | Computing Relational Sufficient Statistics for Large Databases |  |  |  | | --- | --- | | **Masked Reviewer ID:** | Assigned\_Reviewer\_21 | | **Review:** |  |  |  |  | | --- | --- | | **Question** |  | | Overall Rating | Weak Reject | | Summary of the paper (what is being proposed and in what context) and a brief justification of your overall recommendation. One paragraph | The gist of this work lies in generating contingency tables (partial counts) of a database for machine learning applications, in particular for Bayesian Net training and testing. The main contribution lies in generating the counts containing \*negative\* examples, without instantiating all of the possible records, since that may be too expensive. However, the technical depth of the contribution is limited in nature: the final setup is relatively simple and the computational bounds (especially those exponential in the number of relations) are relatively weak. | | Three (or more) strong points about the paper (Please be precise and explicit; clearly explain the value and nature of the contribution). | - Good motivation (data preparation for Bayes Nets learning)  - Interesting question | | Three (or more) weak points about the paper (Please indicate clearly whether the paper has any mistakes, missing related work, or results that cannot be considered a contribution; write it so that the authors can understand what are seen as negative aspects | - Poor theoretical performance guarantees  - Lack of comparison against state of the art methods (are Bayes nets the right way to attack this problem) | | Relevant for PVLDB | YES | | Novelty (Please give a high novelty ranking to papers on new topics, opening new fields, or proposing truly new ideas; assign medium ratings for delta papers and papers on well known topics but still with some valuable contribution). | Novelty unclear | | Significance | Improvement over existing work | | Technical Depth and Quality of Content | Syntactically complete but with limited contribution | | Experiments | OK, but certain claims are not covered by the experiments | | Presentation | Excellent: careful, logical, elegant, easy to understand | | Detailed Evaluation (Contribution, Pros/Cons, Errors); please number each point | The main contribution of this work lies in counting sufficient statistics in contingency cells with negative relations in a manner that never constructs non existing tuples, just counts them. The algebra behind the counting is relatively straightforward - and works by subtracting the number of tuples satisfying a particular condition from the total possible number of tuples. Special care is taken in making sure that this is done efficiently. I am not sure the significance of the work meets the bar for VLDB.   From a machine learning standpoint, I found the limitation to only evaluate Bayes Nets approaches limiting, since it is not clear to me that these are the best for the datasets in question. (Do other ML techniques perform even better?) Although it is not the focus of the paper, it is hard to judge the significance of the contribution without an apt baseline. Namely, do the algorithms in this work improve on the best known learning approaches for the problems in question? |  |  |  | | --- | --- | | **Masked Reviewer ID:** | Assigned\_Reviewer\_7 | | **Review:** |  |  |  |  | | --- | --- | | **Question** |  | | Overall Rating | Neutral | | Summary of the paper (what is being proposed and in what context) and a brief justification of your overall recommendation. One paragraph | The paper presents an algorithm to compute cross-table sufficient statistics of column combinations that are present or absent in the database. The algorithm works by computing the statistics of smaller combinations first, and using dynamic programming to compute the statistics of more complex combinations from the smaller combinations. | | Three (or more) strong points about the paper (Please be precise and explicit; clearly explain the value and nature of the contribution). | 1. Computes the statistics of the so-called 'negative relationships', i.e. combination of column values that are not seen in the database.   2. The algorithm only performs limited materialization, and is therefore orders of magnitude faster than baselines that do. | | Three (or more) weak points about the paper (Please indicate clearly whether the paper has any mistakes, missing related work, or results that cannot be considered a contribution; write it so that the authors can understand what are seen as negative aspects | 1. The presentation of the paper needs a bit of work:  a) What is the schema of the random variable database? In particular I found the entries in Figure 6 hard to understand. What is rnid? What columns are present in RV.RNodes\_pvars?   b) The description of the Lattice Computation subsection (Page 6) is unclear. I think this part of the paper definitely needs to be illustrated with a non-trivial running example.   There are other minor mistakes in the paper, that I have listed in Q14 below.   2. The algorithm uses the cross-product as one of its base case operations. This is expensive if both the tables have >=1 attributes with a large number of distinct values. It also limits the algorithm to small-medium sized databases, as such a cross product would be prohibitive in large databases. The biggest database used in the experiment still has only O(1M) tuples. The paper does not report any empirical scalability results of running the algorithm on a large database (perhaps synthetically generated).   3. The sufficient statistics need to be updated in the case of record insertions/deletions. It seems to me that the entire dynamic programming chart needs to be recomputed if one of the leaf tables is modified. Is that indeed the case, or can something more clever be done to update the statistics incrementally? This is not discussed in the paper.   4. Section 7 needs to be described much better. The paper claims that their Learn-and-Join algorithm learns both the structure and the parameters of the bayesian net, but does not properly describe how the structure is learnt. In particular:  a) What does it mean to "propagate learned edges to chains of length l+1" (page 8)?  b) How does the structure get enhanced as we move to more complex relationship chains?  c) Who provides the edge constraints? A domain expert? How different are the Bayes nets if no constraints are incorporated?    5. The paper claims capturing -ve relationships as one key contribution of the paper,  but does not show that learning a Bayes net that exploits them is better than a bayes net that does not. The paper stops short at learning a bayes net and comparing its complexity etc, but does not compare its effectiveness (e.g. for prediction problems). This would have been a really compelling selling point of the paper but wasn't explored. | | Relevant for PVLDB | YES | | Novelty (Please give a high novelty ranking to papers on new topics, opening new fields, or proposing truly new ideas; assign medium ratings for delta papers and papers on well known topics but still with some valuable contribution). | With some new ideas | | Significance | Improvement over existing work | | Technical Depth and Quality of Content | Syntactically complete but with limited contribution | | Experiments | OK, but certain claims are not covered by the experiments | | Presentation | Reasonable: improvements needed | | Detailed Evaluation (Contribution, Pros/Cons, Errors); please number each point | 1. Table 1: I think 'teachingability' should be 'capability'   2. Algorithm 1: R should be equal to (R\_1,...,R\_l) instead of (R\_pivot, ..., R\_l).   3. Page 7: Not all ct\_ops are equally expensive. Some like cross-product can have a high variance in costs, yet all ct\_ops are counted the same. Is this standard practice?   4. The subsection on dataset description (page 9, col 1) can be dropped as Table 2 is sufficient for the job. The resulting space can possibly be used to illustrate lattice computation (algorithm 2).   5. Sec 9 mentions that the runtime over IMDB was 2 hours. What was the break up? If cross-products take up a lot of this time, then how would one scale to bigger databases / more tables?   6. Page 10: It says that LAJ is much faster on Hepatitis and MovieLens, but Table 4 shows that LAJ is slower than Flat on MovieLens.   7. Page 10: "Fat search" should be "Flat search".    Page 11: Some of the previous works seem quite relevant for specific parts of the paper, yet no empirical comparison is done against them. The only comparison is against a rather naive baseline. E.g. [6] is relevant if one has only 1 -ve relationship, so it would have been good to compare against [6] in that scenario. Or, show that [6] does a poor job of learning a bayes net if there are >1 -ve relationships (see the last 'weak point'). |  |  |  | | --- | --- | | **Masked Reviewer ID:** | Assigned\_Reviewer\_9 | | **Review:** |  |  |  |  | | --- | --- | | **Question** |  | | Overall Rating | Neutral | | Summary of the paper (what is being proposed and in what context) and a brief justification of your overall recommendation. One paragraph | The authors study the efficient computation of sufficient statistics from large relational databases. Sufficient statistics are stored in contingency tables. The authors introduce algebra to enable efficient manipulation of contingency tables. A virtual join algorithm is proposed to compute the contingency table for the entire database in a bottom-up manner. An application of the algorithm to Bayes net learning is studied and tested empirically. | | Three (or more) strong points about the paper (Please be precise and explicit; clearly explain the value and nature of the contribution). | 1. The paper studies the caching of sufficient statistics that may enable efficient machine learning on relational databases.  2. The paper is well written. Examples are provided to illustrate the main idea. | | Three (or more) weak points about the paper (Please indicate clearly whether the paper has any mistakes, missing related work, or results that cannot be considered a contribution; write it so that the authors can understand what are seen as negative aspects | 1. The limitations of the proposed technique should be made clearer.  2. Space complexity of the algorithm is missing.  3. Experimental study can be improved. | | Relevant for PVLDB | YES | | Novelty (Please give a high novelty ranking to papers on new topics, opening new fields, or proposing truly new ideas; assign medium ratings for delta papers and papers on well known topics but still with some valuable contribution). | With some new ideas | | Significance | Improvement over existing work | | Technical Depth and Quality of Content | Syntactically complete but with limited contribution | | Experiments | OK, but certain claims are not covered by the experiments | | Presentation | Reasonable: improvements needed | | Detailed Evaluation (Contribution, Pros/Cons, Errors); please number each point | 1. The computation of full contingency table is applicable only if the number of variables is small and their domain sizes are small. This should be made clearer in the paper. There is in fact a tradeoff between the marginalization of a full contingency table and the joining of small contingency tables. In what circumstances the marginalization of a cached contingency table is better? Making this clearer can better position the paper.   2. The space complexity of the proposed algorithm should be discussed. This is essential to assess the applicability of the algorithm.   3. In experiments, the domain size of each variable is not reported. In learning Bayes net, it is unclear whether the full contingency table is used only or other small ones are used as well. The experiments should also study the tradeoff between the marginalization of a full contingency table and the joining of small contingency tables. It should also compare with the approach of joining relational tables on demand.   4. As mentioned in conclusion, the proposed approach may not scale well to large number of variables (a typical case in relational databases), while the biggest challenge in Bayes net learning mainly comes from the large number of variables. In this sense, Bayes net learning may not serve as a good application unless in a very restricted setting. Better applications of the proposed approach are expected. | |  |