Limiting Data Exposure in Multi-Label Classification Processes

Nicolas Anciaux[1,2], Danae Boutara[1,4], Benjamin Nguyen[1,2], and Michalis Vazirgiannis[3,4]

[1] INRIA, SMIS team, Domaine de Voluceau, 78153 Le Chesnay, France

[2] Université de Versailles St Quentin, 45 av. Etats-Unis, 78035 Versailles, France

[3] Athens U. of Economics & Business, Patision Street, Athens, Greece

[4] LIX, Ecole Polytechnique, 91128, Palaiseau, France

Abstract. Administrative services such social care, tax reduction, and many others using complex decision processes, request individuals to provide large amounts of private data items, in order to calibrate their proposal to the specific situation of the applicant. This data is subsequently processed and stored by the organization. However, all the requested information is not needed to reach the same decision. We have recently proposed an approach, termed *Minimum Exposure*, to reduce the quantity of information provided by the users, in order to protect her privacy, reduce processing costs for the organization, and financial lost in the case of a data breach. In this paper, we address the case of decision making processes based on sets of classifiers, typically multi-label classifiers. We propose a practical implementation using state of the art multi-label classifiers, and analyze the effectiveness of our solution on several real multi-label data sets.

1 Introduction

When an individual completes an administrative procedure (e.g., requesting social care, paying taxes, contracting a loan, etc.) she must usually provide personal information, often requested through an application form. Based on that information, the organization launches a decision process, and determines the set of benefits that must be granted to the applicant (e.g., set of welfare benefits, tax exemptions, features of a loan, etc.). When the decision process is complex (e.g., identifying social needs, tax returns, loans characteristics, etc.) up to thousands of personal data items may be requested.

Decisions are mostly taken automatically, using classifiers made of logical rules established by experts (based on existing laws and directives) or generated by data mining tools. Each potential benefit is thus formalized by a logical rule. For example, a dependent person requesting social assistance may benefit from financial support for a home aid in the following cases: having (i) a pension under

€30.000 and an age above 80, (ii) a pension under €10.000 regardless of age, or (iii) more than two lost abilities (e.g., dressing and bathing independently). This rule is a Boolean Disjunctive Normal Form formula leading to the *home_aid* benefit. We call it *collection rule*, composed of three *atomic rules*, each composed of one or more *predicates*, as illustrated in Figure 1.

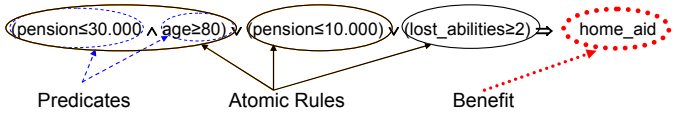

Fig. 1 An Example Collection Rule.

More formally, sets of potential benefits are modeled as sets of collection rules which form a *Multi-Label Classifier* (each benefit being equivalent to a label). The input of this classifier is the union of all data items possibly involved in the multi-label classifier, i.e., involved in a given predicate of a collection rule.

In this context, the questions we address in this paper are the following:

(1) *How to improve the privacy of the applicants?* Personal data items are collected, accessed and processed by organizations workers, and stored afterwards for long durations (even rejected applications may be kept for years for auditing or in the case of dispute). Privacy principles such as Limited Data Collection and Retention, recognized in privacy laws worldwide[1] and implemented in privacy aware computing systems [1], must be achieved.

(2) *How to decrease data processing cost?* The processing of each application induces manual operations. Organisations workers usually check the truthfulness of the data items provided by the applicant, e.g., by verifying certificates, cross checking internal databases or scrutinizing the adequacy of the provided data with respect to attached documents (e.g., copy of the tax returns). While using digital signatures helps alleviating manual checking costs, an important part of the data is issued from unsigned documents or by statements by the applicant.

(3) *How to limit financial loss in the case of a data breach?* Data leaks are now considered a serious threat by organizations. A recent study [25] estimates the cost for US companies at an average $194 per lost record (with an average $5.5million per incident). This prohibitive cost is partially due to negative publicity, but mainly linked to the (recent) legal obligation enacted in many countries (most US states and Europe) to notify and assist the victims in minimizing the impact of the breach (e.g., cancelling a credit card if its number has been disclosed).

We have proposed in [2, 3] a new approach called *Minimum Exposure* to drastically reduce the set of data items collected from applicants, while preserving the same final decision. Considering appropriate metrics capturing privacy considerations, financial costs, or both, a leap forward towards a solution to the above questions can be achieved. Data minimization is a complex task. Let us consider the collection rule pictured in Figure 1. Organizations would request the

[1] See founding privacy laws like the EU Directive 95/46/EC on the protection of individuals with regard to the processing of personal data, and the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data.

union of all data items involved, namely [*pension*, *age*, *lost_abilities*], while only a subset of them is actually required depending on the applicant's situation. Data minimization can be achieved by producing a set of *collection rules* formalizing the decision process, and using an algorithm to automatically expunge useless data items provided by applicants, as proposed in our previous works [2, 3].

The precise goal of this paper is to apply this approach in the context of real multi-label classification. More precisely, the contribution is twofold: (i) we propose a new architecture adapted to the context of multi-label classifiers, based on the *Minimum Exposure* approach, and (ii) we propose an experimental platform, able to transform any real multi-label datasets into collection rules and to measure the gain obtained in terms of data exposure, to validate the approach.

The outline is as follows: Section 2 discusses related works; Section 3 proposes limited data exposure architecture for multi-label classifiers; Section 4 introduces our experimental platform; Section 5 presents measures; and Section 6 concludes.

2 Related Works

In this section, we discuss the related works and then give a short technical background on multi-label classification, as required to introduce our contribution.

Limited data exposure in computing systems. Privacy aware computing systems already address the problem of limiting the data exposure. Examples include the P3P Platform [11], policy languages like EPAL [7], XACML [14] or WSPL [5], and Hippocratic databases [1]. P3P highlights conflicting policies, and thus enable users to avoid accessing services considered as too invasive, but it offers no mean to minimize the data exposed by a user. Hippocratic databases [1] address the legal principle of Limited Data Collection by maintaining for each purpose, the set of required attributes. However useful attributes are derived statically from purposes, without considering the values and combinations of those attributes. This may hold for simple cases (e.g., the *address* is required to deliver a product), but not in multi-label classifiers, more complex by nature (see Figure 1).

Dimensionality reduction in data mining aims at reducing the number of variables taken into consideration in classification, and as such may be considered as a way to limit data exposure. This is usually achieved for clustering algorithms using techniques such as principal component analysis or factor analysis [16]. However, the main difference with our work is that the new space constructed has base vectors which are linear combinations of initial ones (e.g., based on eigenvectors of greatest eigenvalue in the covariance matrix). In our case, it will be impossible to *check* their authenticity, which makes these techniques unusable.

Existing works closer to our study was conducted in the area of automated trust negotiation where access decisions are granted after evaluating credentials requests. For each request the minimum set of credentials is disclosed. Some previous works specifically address this minimization step [6, 10, 21]. However, proposed solutions can not be used here because (i) we consider multi-label

decisions [19] where several benefits are considered (e.g., provide human support, material assistance, home improvement, etc.) rather than a binary access decision, and (ii) we consider large amounts of personal data items (e.g., hundreds to thousands) rather than few credentials (e.g., [6] scales up to 35 data items only).

Multi-Label classification. The specificity of multi-label classification is to consider that several labels must be assigned to each user's application instance. Many organizations are classifying users' applications on several different dimensions, since the final decision is taken considering these dimensions. For example, our partner the General Council of Yvelynes District in France, which is responsible for allocating social care in this district, uses one classifier –binary or single-class– for each potential social benefits (among more than 50) that can be allocated to applicants, e.g., installation of some specialized equipment, providing human assistance in common activities (dressing, bathing, eating, etc.), offering financial support to achieve particular purposes, making home adaptation, offering transportation facilities, etc. All these classifiers considered together form a multi-label classifier. This case is also encountered in many other contexts, like in tax exemption scenarios (one classifier per possible tax reduction), or when contracting insurance and bank loans (many parameters of the bank loan can be adjusted to the specific situation of the applicant, e.g., amount, rate, duration of the loan, job insurance discount, etc.).

In the recent area of multi-label classification, two main methods are proposed to build a classifier from a multi-label dataset: *Problem Transformation* (PT) methods and *Algorithm Adaptation* methods. PT methods transform the multi-label problem into a set of binary classification problems, and then any classification algorithm proposed for binary or multi-class classification can be used. Algorithm adaptation methods adapt the algorithms to directly perform multi-label classification. In this study, we will use existing PT techniques to obtain multi-label classifiers. The PT methods that we use are PT3 and PT4, as a recent study [19] points them out as having the best results among all other PT algorithms. PT3 takes as input a given multi-label dataset with $|L|$ different labels and transforms it to a single-label output dataset obtained by merging the $|L|$ different labels into one. The PT4 method takes the same input as PT3, but instead of producing a single output file, it generates $|L|$ files, each of them containing one label of the original dataset. So, the output here consists of |L| single-label files.

3 Limited Data Exposure Architecture for Multi-Label Classifiers

On Figure 2, we present (in grey font) the main steps that are usually undertaken in usual application evaluation processes based on a multi-label classifier: (1) the *Applicant* (A) retrieves the application form from the *Organization* (Org), she fills that application out according to personal documents

she owns (which may include signed and unsigned information) and returns it; (2) the organization checks the validity of each provided data item (e.g., by checking certificates automatically and by performing manual checking for unsigned data items) and submits the application to a multi-label classifier to determine the *Proposal* (answer) that can be made to the applicant. The processing information (including the application form) is usually (3) stored in a database, possibly for several years, for later use (e.g., social organisations as well as banks must be able to prove that they calibrate their offers using non discriminative legal criterion).

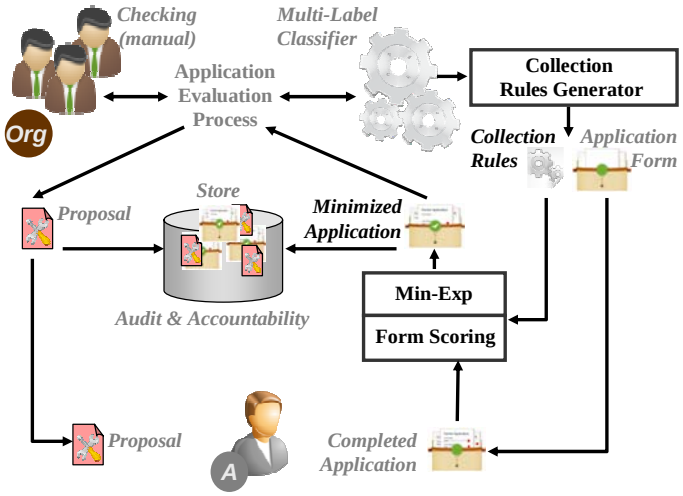

Fig. 2. Limited Data Exposure Architecture.

We introduce three modules to this classical architecture (see the elements in black font on Figure 2): *Collection Rules Generator*; *Form Scoring*; and *Min-Exp* module, used to minimize the set of data items contained in the user's application before it is processed and stored by the organization.

The ***Collection Rules Generator*** produces the *collection rules* based on the multi-label classifier. Those rules are subsequently used by the *Min-Exp* module to determine the minimum set of data items which effectively impacts the classifier. In many cases, classifiers are natively rule based, because rules are produced manually (e.g., derived from laws), and thus the translation into collection rules is obvious. In other contexts, the classifiers are made of black box data mining tools such as, e.g., neural networks or support vector machines. In that case, existing algorithms [8, 13] can transform them into sets of collection rules. The set of data items to be collected from the applicant, and used to feed the multi-label classifier, can be derived directly from the collection rules, by making the union of the attributes involved in the collection rules predicates.

The ***Form Scoring*** module binds a score (the value returned by a *Cost Function*) to each data item entered in the user's application. The difficulty resides in finding good metrics to capture the different aspects: privacy for the applicant, and processing or breach costs for the service provider. Traditional information loss metrics like minimal distortion [17, 18] or ILoss [20] can be considered good

candidates functions. Indeed, since these metrics were created and used for privacy preservation, they accurately measure privacy. However they might also be accurate for measuring processing costs: it is obvious the (manual) checking cost for the organization depends on the volume of processed data. Moreover, the overhead induced by the cost of a data breach is also proportional to the amount of exposed data, as shown by a recent study [15]. Thus, we can find reasonable candidate cost functions considering organizations' costs and users' privacy are both tightly linked to information loss. Our proposal can however accommodate any metric that associates an exposure value to each item independently (e.g., the aforementioned metrics, values entered by a user representing her privacy perception, values represented the checking cost for an organization measured in human minutes by experts, etc.).

The role of the ***MinExp*** module is to compute the benefits that the applicant can receive, if she exposed *all* the data requested in the application form, and then to reduce the amount of data to be exposed (minimized application), while receiving the same benefits. The module is parameterized by the organization *collection rules* on the one hand, and *cost function* on the other. Each collection rules predicate is a Boolean variable corresponding to one of the applicants' attributes, such as age, income, average blood pressure, etc.

We have formally stated this problem in our previous work [3], and we have shown that it is an extension of the Min-Weighted Satisfiability optimization problem, and therefore NP-Hard. Obviously, due to the hardness of the problem, an exact resolution using a Binary Integer Program solver may require important processing time, increasing exponentially with the size of the problem. In practice, approximate resolution such as RAND* as proposed in [3, 4] may be needed, depending on the topology of the collection rules, when the number of data items considered in the application is over one or two hundreds (see Section 5).

4 Experimental Framework

To validate our proposal, we build a platform which takes in input any multi-label dataset, produces a set of collection rules from that dataset, and then measures the gain obtained using *Min-Exp* for each instance in that dataset.

The terminology that we use is the following:

- A **Predicate** is an expression of the form $attribute \theta value$, where *attribute* is an attribute and *value* is its value and where θ is a comparator in $\{=, <, >, \leq, \geq, \neq\}$. In the example of Figure 1, $lost\_abilities \geq 2$ is a predicate.
- An **Application** is the set of data items contained in a user application, being represented as a set of $attribute = value$ predicates where *attribute* is the type of data item (e.g., *pension* and *lost_abilities*, in Figure 1) and *value* is its value.
- A **Label** (or *application label*) is associated to a given application, and represents a benefit granted to the applicant (e.g., the financial support for a *home_aid* in the example of Figure 1).

- A **Multi-Label Dataset** is set of *applications* with associated *labels*, denoted by {<*application*, {*label*}>}, where each application is related to a set of labels. We note $|L|$ the number of distinct labels in the dataset.
- A **Single-Class Dataset** is a set of <*application, label*>, where each *application* is associated with a single *label*, called a *class* in the traditional classification terminology.
- An **Atomic Rule** is a conjunction of *predicates* which leads to a given *label*. In the example of Figure 1, (*pension* $\leq$ *30.000* $\wedge$ *age* $\geq$ *80*) is an atomic rule leading to label *home_aid*.
- A **Collection Rule** is the disjunction of all the *atomic rules* leading to a given *label*. In the example of Figure 1, ((*pension* $\leq$ *30.000* $\wedge$ *age* $\geq$ *80*) $\vee$ (*pension* $\geq$ 1*0.000*) $\vee$ (*lost_abilities* $\geq$ *2*)) $\rightarrow$ *home_aid* is a *collection rule*.
- A **Cost Function** is a function representing the privacy/processing/breach cost of a set of predicates. In this paper, we use a simple cost function that counts the number of distinct attributes involved in those predicates (i.e., the number of data items exposed by a user in her application).
- The **Full Graph** is the bipartite graph representation of a *complete* set of collection rules. Sets of collection rules are represented as the bipartite graph $G = (P \cup L, R, E_P \cup E_L)$ where *P, R, L* are respectively the sets of *predicates*, *atomic rules*, and *labels*, involved in the collection rules, and where E_F is the set of vertices between *P* and *R* with the interpretation "$e=(p\in P, r\in R)\in E_F \Leftrightarrow p\in r$" meaning that predicate *p* is involved in the atomic rule *r*, and E_L the set of vertices between *L* and *R*, with the interpretation "$e=(l\in L, r\in R)\in E_L \Leftrightarrow (r\Rightarrow l)$" meaning that atomic rule *r* leads to label *l*.
- The **Local Graph** is a subset of the *full graph* obtained for a given application, i.e., the graph obtained by removing from the full graph all nodes (predicates, labels and atomic rules) and all edges that cannot be satisfied when considering the content of that given application.
- A **Minimized Graph** ($P_m \cup L_m$, R_m, $E_{Pm} \cup E_{Lm}$) can be constructed from both full and local graphs, given a cost function *c*. A minimized graph is such that $L_m=L$, $R_m\subseteq R$, $E_{Pm}\subseteq E_P$, $E_{Lm}\subseteq E_L$, $P_m\subseteq P$ and $c(P_m)$ is minimal.

From a multi-label dataset, we build (full and local) Graphs used as inputs of the module computing Minimized Graphs. This is done in 4 steps (see Figure 3):

1. ***Problem Transformation (PT).*** It implements the PT classification algorithms called PT3 and PT4 (see Section 2) transforming the multi-label datasets into single-label ones. For a given multi-label dataset with $|L|$ labels used in input, $|L|$ single-class datasets are produced in output.
2. ***Single-Label Classification.*** Traditional classification algorithms can be used to classify each single-class dataset. We have chosen *RIPPER* [10], a state of the art classifier implemented for the *WEKA*[2] framework (*Weka.JRIP* class). For each single-class dataset {<*application, label*>}, RIPPER produces a set of

[2] See http://sourceforge.net/projects/weka/

atomic rules leading to that *label*. All single-class datasets are consumed and the resulting atomic rules are dumped into two (CSV) files, the first of which contains the collection rules and the second some statistics about those rules (e.g., coverage/uncoverage, true/false positives/negatives).

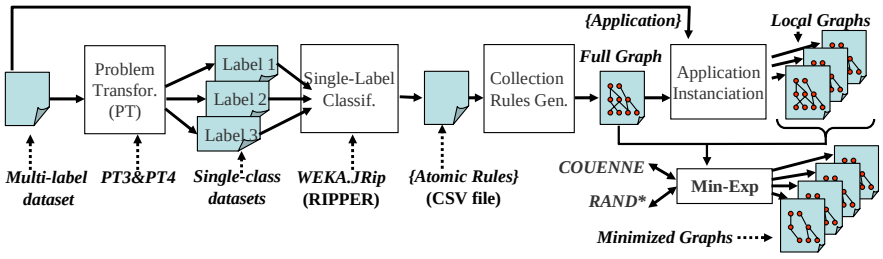

Fig. 3. Experimental Platform.

3. ***Collection Rules Generator (produces Full Graph).*** Only the most interesting atomic rules (according to statistics) are selected to be further considered. By constructing the union of all the atomic rules leading to a given label, we obtain a collection rule. The conjunction of all the collection rules forms the multi-label classifier, represented as a bipartite graph (see the notations above). The result of these steps is called the *full graph*, which corresponds to a bipartite graph representation of the complete set of collection rules. This structure is then used as an input of the *Min-Exp* module. The computation of the Minimum Exposure on this full graph corresponds to the computation made for a *virtual* application which would contain enough data items to satisfy all the predicates in the collection rules.
4. ***Application Instantiation (produces Local Graphs).*** We compute for each application the *local graph*, which is a sub-graph of the full graph. It is done by removing any node/edge in the full graph that cannot be satisfied given the predicates composing the considered application. The local graphs are then used as inputs of the *Min-Exp* module which produces the *minimized graphs*.

The problem of computing the minimized graph, that we call *Minimum Exposure problem* is *NP-hard*, as shown in [3]. The problem can be solved with any state of the art Mixed Integer Non-Linear Programming (MINLP) solver and with approximate algorithms (since the solver may consume far too much time when the size of the problem increases). We sketch below the resolution techniques used in our experiments, and refer the reader to [3] for details.

COUENNE. We used *COUENNE* (Convex Over and Under ENvelopes for Nonlinear Estimation) as a representative state of the art solver [9]. As *COUENNE* is spatial branch-and-bound, its worst-case complexity is *exponential* in the number of variables, both integer and continuous. To generate an instance of a problem solved by a MINLP solver, we used AMPL, an algebraic modelling language for optimization problems on discrete or continuous variables. In

practice, COUENNE finds an optimal solution in reasonable time in most cases but not for large instances, as shown in Section 5.

RAND*. We used a simple approximation algorithm called *RAND** to serve as a baseline algorithm and to obtain approximate solutions when the solver fails finding any optimal one. From an input full or local graph, *RAND** randomly chooses one atomic rule in each collection rule, and considers the union of the *attributes* involved in those atomic rules as a potential result. As all labels are covered, this set of attributes corresponds to a solution. The operation is repeated, and the result minimizing the exposure (using the chosen exposure metric) is kept.

5 Measures

Experiments were conducted on a HP workstation with 3.1GHz Intel CPU and 8GB RAM running Java1.6 (x64). *COUENNE* was run on the same machine.

In our experiments, we have run *COUENNE* and *RAND** on full and local graphs obtained using three real datasets. The first two datasets are named *ENRON* and *MEDICAL*, and are publicly available from the MULAN website[3]. The third one is called *SOCIAL*, and was build with the help of our partner the General Council of Yvelines District. Those datasets are as follows:

- ***ENRON*.** This dataset contains e-mails of the Enron employees, made public in 2003 by the Federal Energy Regulatory Commission. It contains 1702 emails (i.e., *applications* with our notations) involving 1001 nominal attributes (email keywords) categorized into 53 different labels.
- ***MEDICAL*.** This dataset was collected from the Cincinnati Children's Hospital Medical Center's Department of Radiology. It contains a sampling of patients' chest x-ray and renal procedures for one year. It contains 978 instances (i.e., applications) and 1449 nominal attributes (medical histories of patients), which are classified in 45 different labels (characterizing diseases of patients).
- ***SOCIAL***. This dataset was constructed with the help of the General Council of Yvelines Distrinct. It contains anonymous samples of application forms sent by dependent people to request social assistance (the main form has 440 different fields). We have used those samples to generate local graphs. Our framework (see Section 4) could not be used to build the corresponding multi-label classifier (the full graph), because not enough samples were available. We did however easily build the corresponding multi-label classifier, with the help of General Council experts, by deriving it from laws and existing General Council directives. It involves 56 labels representing the potential dimensions of social help provided to applicants (e.g., provision of specialized equipment, human assistance for dressing, bathing or eating, financial assistance, home adaptation, transportation facilities, etc.).

[3] http://mulan.sourceforge.net/datasets.html

The topology of the full graphs is presented in Table 1. For *ENRON* and *MEDICAL* datasets, the graphs were generated using our framework (presented in Section 4), and for *SOCIAL* it was produced with the help of experts.

Graph / *Dataset*	*Predicates*	*Atomic Rules*	*Labels*
ENRON	122	140	53
MEDICAL	225	195	45
SOCIAL	440	225	56

Table 1. Full Graphs Topology.

To compare the efficiency of the resolution techniques, we measured for full and local graphs, the data exposure gain: $EXP = (p - p_{MIN}) / p$, where p denotes the number of predicates in the Graph and p_{MIN} denotes the number of different attributes involved in those predicates that was kept by the *Min-Exp* resolution algorithm. This gain is given in function of the execution time. For the approximation algorithm *RAND**, the execution time varies according to the number of iterations allocated to the algorithm (see Section 4), and the maximum time considered is the one taken by *COUENNE* which produced the exact solution. We show the relation between the execution time (x-axis) and the exposure gain (y-axis) for the full graph (Figures 4 to 6), local graphs (Figures 7 to 9, which give the average results), and for the 10% largest *applications*, i.e., with the highest number of distinct data items (Figures 10 and 11).

The main conclusions are the following: (i) the gain (exposure ratio) is always important, above 40% in all our measures; (ii) *RAND** performs relatively well considering that it is a random approximate algorithm; (iii) *COUENNE* gives, as expected, better results than *RAND**; and (iv) for *COUENNE*, the execution time increases exponentially as the size and complexity of the problem grows (for SOCIAL, 1 hour in average was needed per application, and largest applications remained unsolved after 12 hours). Thus *RAND** which provides rather satisfying results, could be used as a replacement of the optimal resolution.

6 Conclusion

In our previous work [3] we have proposed the Minimum Exposure approach to limit data collection in online forms. In this paper, we apply this approach in the context of real multi-label classification. We have adapted the architecture, and have proposed an evaluation platform able to take in input any real multi-label datasets to evaluate the impact on data exposure reduction in real cases. We show that in real cases, the exposure reduction is above 40%. This increases the user's privacy, and minimizes applications' checking costs and financial losses suffered by the organization in the event of a data breach.

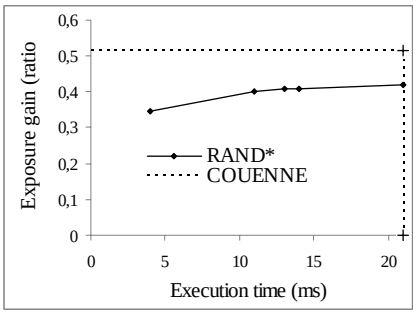

Fig.4. Full Graph - ENRON

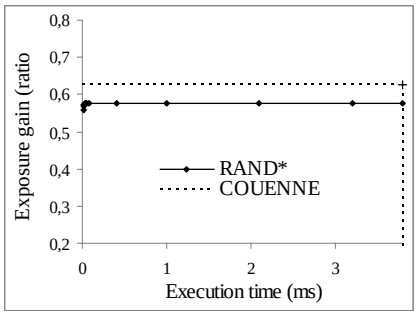

Fig.7. Local Graphs - ENRON

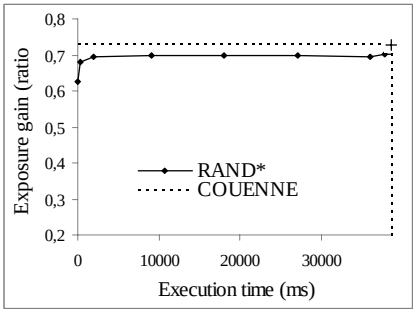

Fig.5. Full Graph - MEDICAL

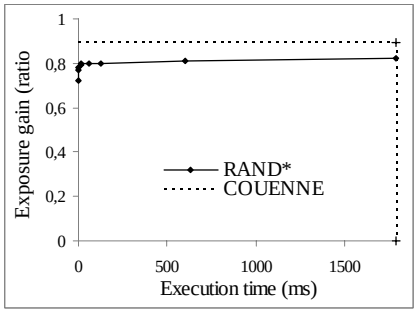

Fig.8. Local Graphs - MEDICAL

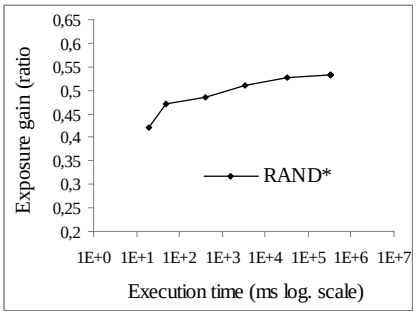

Fig.6. Full Graph - SOCIAL

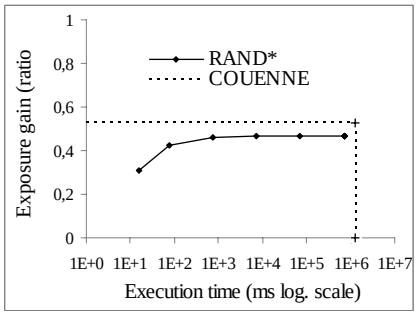

Fig.9. Local Graphs – SOCIAL

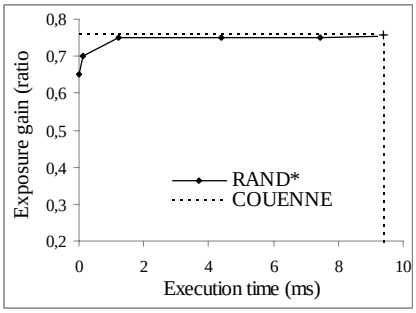

Fig.10. Largest Applications -ENRON

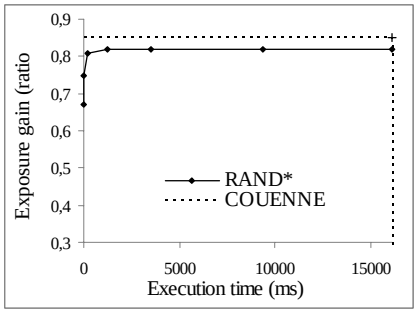

Fig.11. Largest Applications – MEDICAL

Acknowledgments

This work is supported by the KISS grant ANR-11-INSE-0005, by the DIGITEO LeTeVoNe grant, and by the INRIA CAPPRIS grant. We thank Grigorios Tsoumakas for a helpful discussion on multi-label datasets.

References

1. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, Hippocratic databases. In *VLDB*, 2002.
2. N. Anciaux, B. Nguyen, and M. Vazirgiannis, The Minimum Exposure Project: Limiting Data Collection in Online Forms. *ERCIM News*, 90, 2012.
3. N. Anciaux, B. Nguyen, and M. Vazirgiannis, Limiting Data Collection in Online Forms, In *IEEE PST*, 2012.
4. N. Anciaux, B. Nguyen, and M. Vazirgiannis, M. Minimum Exposure in classification scenarios. INRIA Research Report, 2012.
5. A.H. Anderson, An Introduction to the Web Services Policy Language (WSPL), In *POLICY Workshop*, 2004.
6. C.A., Ardagna, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, and P. Samarati, Minimising Disclosure of Client Information in Credential-Based Interactions. *Int. Journal of Information Privacy, Security and Integrity*, *1*(2-3), 2012.
7. P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, Enterprise privacy authorization language 1.2 (EPAL 1.2), W3C Member Submission, 2003.
8. B. Baesens, R. Setiono, C. Mues, J. Vanthienen, Using neural network rule extraction and decision tables for credit-risk evaluation, *Management Science*, *49*(3), 2003.
9. P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wachter, Branching and bounds tightening techniques for non-convex MINLP, *Optim. Methods and Software, 24*(4-5), 2009.
10. W. Chen, L. Clarke, J. Kurose, and D. Towsley, Optimizing cost-sensitive trust-negotiation protocols, In *IEEE INFOCOM*, 2005.
11. L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, *W3C Recomm.*, 2002.
12. R. Fourer, D.M. Gay, and B.W. Kernighan. A Modeling Language for Mathematical Programming. *Management Science*, *36*, 1990.
13. J. Huysmans, B. Baesens, and J. Vanthienen, Using rule extraction to improve the comprehensibility of predictive models, *Open Access publications from Katholieke Universiteit Leuven*, 2007.
14. T. Moses, Extensible access control markup language (xacml) version 2.0. *Oasis Standard*, 2005.
15. Ponemon Institute, LLC. 2010 Annual Study: U.S. Cost of a Data Breach. 2011.
16. R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, John Wiley and Sons Inc, ISBN : 978-0471056690, 2001.
17. P. Samarati, Protecting respondents' identities in microdata release. *IEEE TKDE, 13*(6), 2001.
18. L. Sweeney. k-Anonymity: a model for protecting privacy. *Int. Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*, 2002.
19. G. Tsoumakas, and I. Katakis, Multi-label classification: An overview, *Int. Journal of Data Warehousing & Mining*, 3(3), 2007.
20. X. Xiao, and Y. Tao, Personalized privacy preservation. In *ACM SIGMOD*, 2006.
21. D. Yao, F.B. Frikken, M.J. Atallah, and R. Tamassia, Private information: To reveal or not to reveal. In *ACM TISSEC*, *12*(1), 2008.