Fundamenta Informaticae DKKD and PARIS Special Issue (2013) 1–18
DOI 10.3233/FI-2012-0000
IOS Press

Limiting Data Exposure in Multi-Label Classification Processes

Nicolas Anciaux
INRIA and University of Versailles SMIS team
Domaine de Voluceau
78153 Le Chesnay, France
nicolas.anciaux@inria.fr

Danae Boutara
INRIA and Ecole Polytechnique
Laboratoire d'Informatique de l'Ecole Polytechnique
91128, Palaiseau, France
dboutara@gmail.com

Benjamin Nguyen
INRIA and University of Versailles SMIS team
Domaine de Voluceau
78153 Le Chesnay, France
benjamin.nguyen@uvsq.fr

Michalis Vazirgiannis
Ecole Polytechnique and Athens University of Economics and Business
Laboratoire d'Informatique de l'Ecole Polytechnique
91128, Palaiseau, France
mvazirg@aueb.gr

Address for correspondence: Dr. Benjamin Nguyen, INRIA, SMIS team, Domaine de Voluceau, 78153 Le Chesnay, France, email : benjamin.nguyen@uvsq.fr

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.

Keywords: Privacy, Online forms, Overdata Disclosure, Limited Data Collection, Multi-label Classification

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.

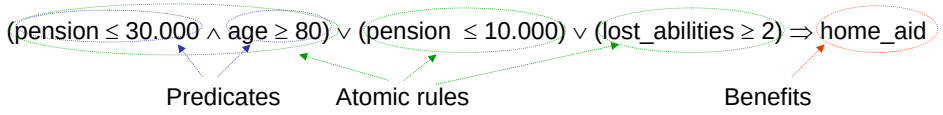


Figure 1. An Example Collection Rule.

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 [3], 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 [6, 4] 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 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 [6, 4].

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 framework and experimental platform; Section 5 presents measures; and Section 6 concludes.

2. Related Works

In this section, we discuss the related works by provinding an overview of how Limited Data Exposure is managed currently. We discuss trust negotiation, which is a topic where the solutions provided are

[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.

close to the ones used in our problem. We also discuss Privacy Preserving Data Mining (PPDM) and Privacy Preserving Data Publishing (PPDP). We then give a short technical background on multi-label classification, as required to introduce our contribution.

2.1. Limited data exposure in computing systems

Privacy aware computing systems already address the problem of limiting the data exposure. Examples include the P3P Platform [14], policy languages like EPAL [9], XACML [24] or WSPL [7], and Hippocratic databases [3]. 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 [3] 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 [15]. 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 [8, 13, 31]. However, proposed solutions can not be used here because (i) we consider multi-label decisions [28] where several benefits are considered (e.g., provide human support, materia 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., [8] scales up to 35 data items only).

2.2. Privacy Preserving Data Mining

Works dealing with Privacy Preserving Data Mining (PPDM) also take a different direction than Minimum Exposure. Recent PPDM surveys [2, 17] refer neither to Minimum Exposure type problems nor to their legal foundation (i.e., the Limited Data Exposure principle). Unlike Minimum Exposure, PPDM techniques, such as recent developments in [19, 23, 22] which protect individual records with regards to the input of a data mining algorithm, turn original data into encrypted or randomly perturbed data, which becomes unverifiable. On the contrary, Minimum Exposure preserves the original data and its ability to be verified by a third party (a signature guarantees its integrity and origin). Another aspect of PPDM techniques is that they try to protect sensitive rules (i.e., the output of a data mining algorithm) by removing raw data [1, 29]. However, these techniques maximize the information retained in the output data set, so long as the private results remain secret, whereas the goal of Minimum Exposure is to minimize it. Note that this approach is nevertheless compatible with Minimum Exposure. Indeed, the former (PPDM) would remove sensitive data upstream and the latter (MinExp) could minimize the remaining

information, thereby achieving better privacy.

2.3. Privacy Preserving Data Publishing

Privacy Preserving Data Publishing (PPDP) [20] is also closely related to the Minimum Exposure problem. Indeed, PPDP focuses on publishing original raw data rather than data mining results or statistics. However, subsequent treatments are not known at the time of data publishing. In Minimum Exposure, the knowledge of these treatments is a prerequisite to identify the minimum subset of data to be exposed. Furthermore, PPDP tries to balance privacy gain and data utility, sometimes with difficulties [12], while Minimum Exposure preserves the full utility of the data (complete set of due benefits are obtained). Some PPDP techniques like [16], closer to statistical databases can be assimilated to (advanced) access control, where statistical data is exposed without revealing individual values.

2.4. 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 [28] 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

3.1. General Architecture

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).

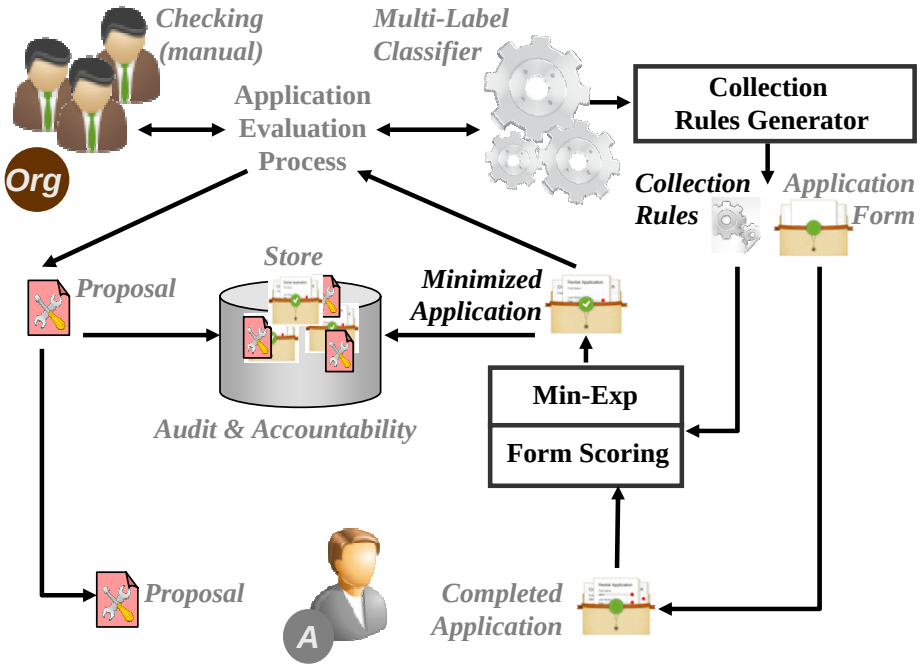


Figure 2. Limited Data Exposure Architecture.

3.2. Modules Description

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 [10, 21] 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 [26, 27] or ILoss [30] 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 [25]. 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.). In our application, it appeared that a very basic cost function (see Definition 4.8) was good enough (in the eyes of the experts) to capture both the processing cost and privacy gain. However, future work includes the study of composite cost functions that combine several existing cost function representing different dimensions. Indeed, one can argue that a specific metric used to represent processing cost, and another to represent information loss could lead to better quality results.

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 [4], 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 [4, 5] 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 4.4 and Section 5).

4. Multi-Label Minimum Exposure Computation Framework

To validate our proposal, we habe built 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 *MinExp* for each instance in that dataset.

4.1. Formalism definitions

We define next the concepts and terminology useful for the rest of this article.

Definition 4.1. (Predicate)
A **Predicate** is an expression of the form $attribute\ \theta\ value$, where $attribute$ is an attribute of the dataset 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.

Definition 4.2. (Application)
An **Application** is the set of data items contained in a user application, being represented as a set of $attribute\ \theta\ value$ predicates where $attribute$ is the type of data item (e.g., $pension$ and $lost\_abilities$, in Figure 1) and $value$ is its value.

Definition 4.3. (Label)
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).

Definition 4.4. (Multi-Label Dataset)
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.

Definition 4.5. (Single-Class 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.

Definition 4.6. (Atomic Rule)
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 $home\_aid$.

Definition 4.7. (Collection Rule)
A **Collection Rule** is the disjunction of all the *atomic rules* leading to a given *label*. In the example of Figure 1, $((pension \leq 30.000 \wedge age \geq 80) \vee (pension \geq 10.000) \vee (lost\_abilities \geq 2)) \rightarrow home\_aid$ is a collection rule.

Definition 4.8. (Cost Function)
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).

Definition 4.9. (Full Graph)
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.

Definition 4.10. (Local Graph)
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.

Definition 4.11. (Minimized Graph)
A *Minimized Graph* $(P_m \cup L_m, R_m, E_{P_m} \cup E_{L_m})$ can be constructed from both full and local graphs, given a cost function c. A minimized graph is such that $L_m = L$, $R_m \subseteq R$, $E_{P_m} \subseteq E_P$, $E_{L_m} \subseteq E_L$, $P_m \subseteq P$ and $c(P_m)$ is minimum.

4.2. Multi-label processing methodology

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). We illustrate these steps by an example in Section 4.3

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* [13], 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 *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).

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 *MinExp* 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.

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

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 *MinExp* module which produces the *minimized graphs*.

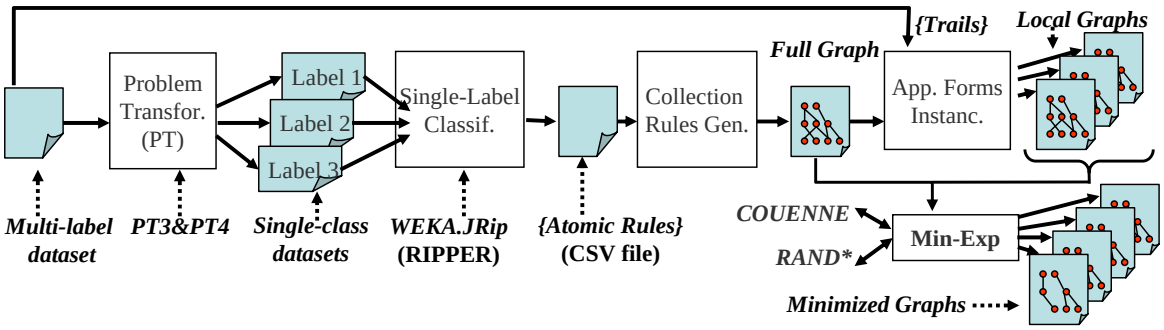


Figure 3. Experimental Platform.

4.3. Example

We give below an example to clarify this process, based on the social assistance use case.

4.3.1. Problem Transformation

The input of step 1 is a sample set of application forms received by the General Council from individuals requesting social assistance. Each complete application contains over 400 items describing the financial ressources of the applicant, her physical, psychological and social abilities, and her living conditions. The sample application forms that we use in this step contain sets of data items, for example[3]: *{age=86, sex=male, living=alone, pension=12.547, read_ability=yes, write_ability=no, shoping_ability=no, clothing_ability=yes, uses_stairs=no, standing = no [and approximately 400 other items]}*

Each application is associated with a set of social measures which were taken after discussions between social workers, medical workers and the authorities of the General Council. The social assistance provided to that applicant is expressed by labels, which for the previous application may include the following:

human_assistance_for_dressing, financial_assistance, bathroom_adaptation, [and potentially up to 53 other labels].

The PT algorithms transform this multi-label dataset into a single label one. We give a very simple multi-labelled dataset, containing only 2 application forms:

$\mathcal{D}$={$Form_1$: <*age=86, sex=male, living=alone, pension=12.547, read_ability=yes, write_ability=no, shoping_ability=no, clothing_ability=yes, uses_stairs=no, standing = no* $\rightarrow$ *human_assistance_*

[3]This example does not correspond to any of the real applications processed for obvious privacy issues, but is provided to illustrate our framework

for_ dressing, financial_ assistance, bathroom_ adaptation>;
$Form_2$ *: <age=79, sex=male, living=alone, pension=35.525, read_ ability=no, write_ ability=no, shoping_ ability=no, clothing_ ability=no, uses_ stairs=no, standing = no* $\rightarrow$ *human_ assistance_ for_ dressing, human_ assistance_ for_ bathing>}*

The outputted single label datasets would be:

- *human_ assistance_ for_ dressing dataset = {<age=86, sex=male, living=alone, pension=12.547, read_ ability=yes, write_ ability=no, shoping_ ability=no, clothing_ ability=yes, uses_ stairs=no, standing = no, YES>; <age=79, sex=male, living=alone, pension=35.525, read_ ability=no, write_ ability=no, shoping_ ability=no, clothing_ ability=no, uses_ stairs=no, standing = no, YES>}*

- *financial_ assistance dataset = {<age=86, sex=male, living=alone, pension=12.547, read_ ability=yes, write_ ability=no, shoping_ ability=no, clothing_ ability=yes, uses_ stairs=no, standing = no, YES>; <age=79, sex=male, living=alone, pension=35.525, read_ ability=no, write_ ability=no, shoping_ ability=no, clothing_ ability=no, uses_ stairs=no, standing = no, NO>}*

- *bathroom_ adaptation dataset = {<age=86, sex=male, living=alone, pension=12.547, read_ ability=yes, write_ ability=no, shoping_ ability=no, clothing_ ability=yes, uses_ stairs=no, standing = no, YES>; <age=79, sex=male, living=alone, pension=35.525, read_ ability=no, write_ ability=no, shoping_ ability=no, clothing_ ability=no, uses_ stairs=no, standing = no, NO>}*

- *human_ assistance_ for_ bathing dataset = {<age=86, sex=male, living=alone, pension=12.547, read_ ability=yes, write_ ability=no, shoping_ ability=no, clothing_ ability=yes, uses_ stairs=no, standing = no, NO>; <age=79, sex=male, living=alone, pension=35.525, read_ ability=no, write_ ability=no, shoping_ ability=no, clothing_ ability=no, uses_ stairs=no, standing = no, YES>}*

4.3.2. Single-Label Classification

Each single label dataset is used to feed the RIPPER algorithm, as implemented in Weka, which produces a set of rules. RIPPER parameters were set to default values[4]. For each rule, a set of statistics is produced. Typically, Jrip gives the rate of true positives, false positives, true negatives, false negatives, coverage and uncoverage for each rule. These statistics are available for experts, to manually discard rules in step 3. With the dataset considered above, the set of rules produced by RIPPER could be :

$$\begin{aligned}
(living = alone \wedge clothing_ability = no) &\rightarrow human_assistance_for_bathing \\
(living = alone \wedge uses_stairs = no) &\rightarrow bathroom_adaptation \\
(living = alone \wedge standing = no) &\rightarrow human_assistance_for_dressing \\
(living = alone \wedge clothing_ability = no) &\rightarrow human_assistance_for_dressing
\end{aligned}$$

[4]The parameters used were: folds=3, minNo=2.0, debug = false, CheckErrRate = true, UsePruning = true

4.3.3. Collection Rules Generator

In certain cases, step 3 is preceded by a manual rule generation step. This step is obviously useless in most cases, because Jrip already selects the most meaningful rules. However, if the set of applications samples is not large enough, Jrip cannot perform accurately. Its outputs (rules and relative statistics) are then only considered as hints helping experts to formulate the rule set manually.

The result of this step is a full graph built from the ruleset. The full graph obtained from the rules expressed in the previous example is shown in Figure 4.

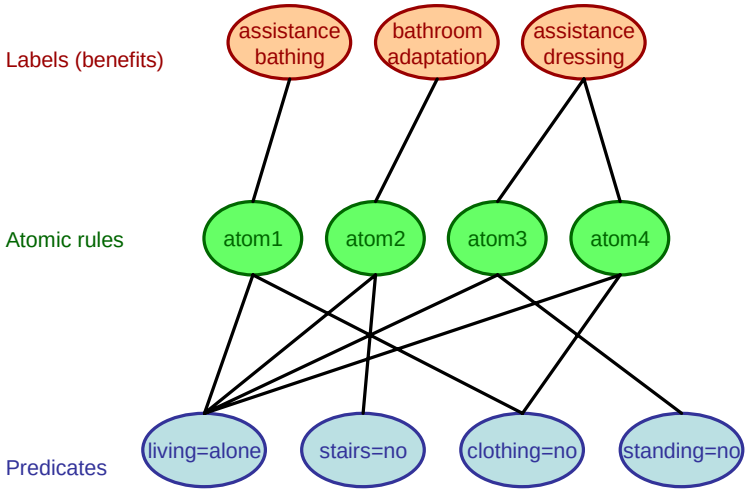


Figure 4. Collection Rules example

4.3.4. Application Instantiation

For each application form, a local graph is derived from this full graph, by removing nodes and edges in the full graph that cannot be activated for that particular application. To illustrate this step, we show in Figure 5 the local graph for the *first* application form considered in the example.

Note that obviously this example, built for sake of illustration, does not (at all) reflect a *real* problem complexity.

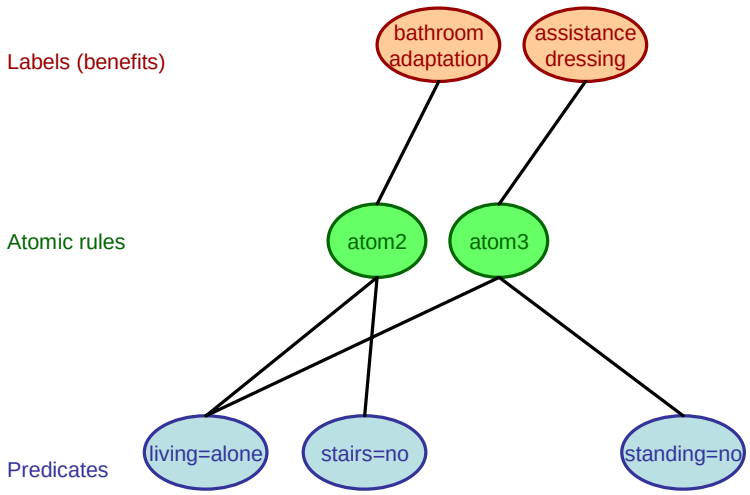


Figure 5. Local graph example for $Form_1$

4.4. Minimization computation

The problem of computing the minimized graph, that we call *Minimum Exposure* problem is NP-hard, as shown in [4]. 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 [4] for details.

COUENNE. We used *COUENNE* (Convex Over and Under ENvelopes for Nonlinear Estimation) as a representative state of the art solver [11]. 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 (see [18]). 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.

The execution time of *COUENNE* is in the worst case exponential in the number of predicates, atomic rules or labels. The execution time of *RAND** depends on the number of random rules generated. Generating one solution depends linearly on the number of labels and the average number of predicates per atomic rule : this is simply the time taken to merge predicates generated (which is linear since atomic rules maintain an ordered list of predicates) and output the solution. In practice, it is very fast, since it takes under $1\mu s$. There are however no theoretical worst case guarantees on the quality (i.e. proximity to the optimal) of the approximation by such an algorithm. In fact, we have shown in [4] that the Minimum Exposure optimization problem is NP-Hard, not in APX[5] and has a differential approximation[6] of 0-DAPX[7]. Therefore for any polynomial approximation algorithm (*a fortiori RAND**), there is no constant upper bound on the quality of the approximation.

Nevertheless, as shown in the following Section in Figures 6(a) to 6(h), the quality of the approximation by *RAND** on real datasets is in practice close to the theoretical optimum (within 10%), even with much less processing time.

5. Experimental setting and results

5.1. Setting

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[8]. The third one is called $SOCIAL$, and was build with the help of our partner the General Council of Yvelines District. Let us now describe these datasets.

- ***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

[5]APX is the class of NP optimization problems that allow polynomial-time approximation algorithms with an approximation ratio bounded by a constant.

[6]Given an instance I of an optimization problem, and a feasible solution S of I, we denote $m(I,S)$ the value of solution $S, opt(I)$ the value of an optimal solution of I and $W(I)$ the value of a worst solution of I. The differential approximation ratio of S is defined by $DR(I,S) = abs((m(I,S)–W(S))/(opt(I)–W(I))$. The traditional approximation ratio for a minimization problem is simply defined by $m(I,S)/opt(I)$.

[7]0-DAPX is the class of NP optimisation problems for which all polynomial approximation algorithms have a differential approximation ratio of 0.

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

graphs. Our framework (see Section 4) alone 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.).

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.

Table 1. Full Graphs Topology

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

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 *MinExp* 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 6(a), 6(c) and 6(e)), local graphs (Figures 6(b), 6(d) and 6(f), which give the average results), and for the 10% largest *applications*, i.e., with the highest number of distinct data items (Figures 6(g) and 6(h)).

The main conclusions are the following:

1. The gain (exposure ratio) is always important, above 40% in all our measures.
2. *RAND** performs relatively well considering that it is a random approximate algorithm.
3. *RAND** scales linearly with the number of labels and average number of predicates per atomic rule, ensuring overall scalability to any real world dataset.
4. *COUENNE* gives, as expected, better results than *RAND**.
5. 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).

In consequence, *RAND** which provides rather satisfying results, could be used as a replacement of the optimal resolution. *RAND** gives the possibility of computing an approximate result in a bounded

amomunt of time, without however having any formal guanratees on the quality of the approximation. We believe that experimental results show that the quality of this approximation is, in practice, quite acceptable.

6. Conclusion

In our previous work [4] 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.

Aknowledgments

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] Aggarwal, C. C., Pei, J., Zhang, B.: On Privacy Preservation against adversarial Data Mining, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[2] Aggarwal, C. C., Yu, P. S.: A general survey of privacy preserving data mining models and algorithms, *Advances in Database Systems*, **34**, 2008.

[3] Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic databases, *Proceedings of the 28th International Conference on Very Large Databases (VLDB)*, 2002.

[4] Anciaux, N., Nguyen, B., Vazirgiannis, M.: Limiting Data Collection in Online Forms, *Proceedings of the IEEE 10th International Conference on Privacy, Security and Trust (PST)*, 2012.

[5] Anciaux, N., Nguyen, B., Vazirgiannis, M.: *Minimum Exposure in classification scenarios*, Technical report, INRIA Rocquencourt, 2012.

[6] Anciaux, N., Nguyen, B., Vazirgiannis, M.: The Minimum Exposure Project: Limiting Data Collection in Online Forms, *ERCIM News*, **90**, 2012.

[7] Anderson, A. H.: An Introduction to the Web Services Policy Language (WSPL), *Proceedings of the IEEE 5th International Workshop on Policies for Distributed Systems and Networks (POLICY)*, 2004.

[8] Ardagna, C. A., de Capitani di Vimercati, S., Foresti, S., Paraboschi, S., Samarati, P.: Minimising Disclosure of Client Information in Credential-Based Interactions, *International Journal of Information Privacy, Security and Integrity*, **1**(2-3), 2012.

[9] Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise privacy authorization language 1.2 (EPAL 1.2), W3C Member Submission, 2003.

[10] Baesens, B., Setiono, R., Mues, C., Vanthienen, J.: Using neural network rule extraction and decision tables for credit-risk evaluation, *Management Science*, **49**(3), 2003.

[11] Belotti, P., Lee, J., Liberti, L., Margot, F., Wachter, A.: Branching and bounds tightening techniques for non-convex MINLP, *Optimization Methods and Software*, **24**(4-5), 2009.

[12] Brickell, J., Schmatikov, V.: The cost of privacy : destruction of Data mining utility in anonymized data publishing, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

[13] Chen, W., Clarke, L., Kurose, J., Towsley, D.: Optimizing cost-sensitive trust-negotiation protocols, *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[14] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J.: The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation, 2002.

[15] Duda, R. O., Hart, P. E., Stork, D.: *Pattern Classification*, John Wiley and Sons Inc, 2001.

[16] Dwork, C., Lei, J.: Differential privacy and robust statistics, *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, 2009.

[17] Evfimievski, A., Grandison, T.: *Privacy Preserving Data Mining*, IGI Global, 2009.

[18] Fourer, R., Gay, D. M., Kernighan, B. W.: *AMPL : A Modeling Language for Mathematical Programming, second edition*, Duxbury Press, 2002.

[19] Friedman, A., Schuster, A.: Data mining with differential privacy, *Proceedings of the 16th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, 2010.

[20] Fung, B. C. M., Wang, K., Chen, R., Yu, P. S.: Privacy Preserving Data Publishing : A Survey, *ACM Computing Surveys*, **42**(4), 2010.

[21] Huysmans, J., Baesens, B., Vanthienen, J.: Using rule extraction to improve the comprehensibility of predictive models, Open Access publications from Katholieke Universiteit Leuven, 2007.

[22] LeFevre, K., DeWitt, D. J., Ramakrishnan, R.: Workload-aware anonymization techniques for large-scale datasets, *ACM Transations on Database Systems (TODS)*, **33**(3), 2008.

[23] Mohammed, N., Chen, R., Fung, B. C. M., Yu, P. S.: Differentially private data release for data mining, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD)*, 2011.

[24] Moses, T.: Extensible access control markup language (XACML) version 2.0., Oasis Standard, 2005.

[25] Ponemon Institute, LLC.: 2010 Annual Study: U.S. Cost of a Data Breach, 2011.

[26] Samarati, P.: Protecting respondents' identities in microdata release, *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, **13**(6), 2001.

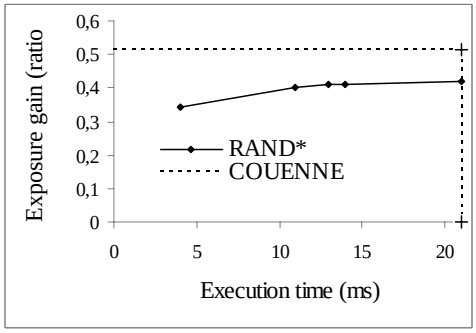
[27] Sweeney, L.: k-Anonymity: a model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, **10**, 2002.

[28] Tsoumakas, G., Katakis, I.: Multi-label classification: An overview, *International Journal of Data Warehousing and Mining*, **3**(3), 2007.

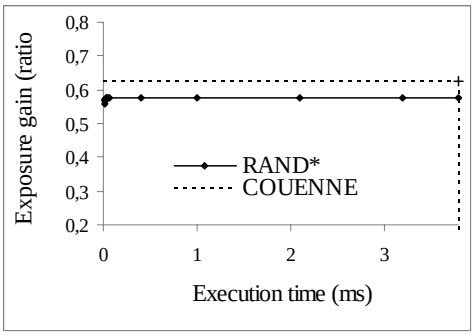
[29] Verykios, V. S., Elagarmid, A. K., Bertino, E., Saygin, Y., Dasseni, E.: Association Rule Hiding, *Transactions on Knowledge and Data Engineering (TKDE)*, **16**(4), 2004.

[30] Xiao, X., Tao, Y.: Personalized privacy preservation, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2006.

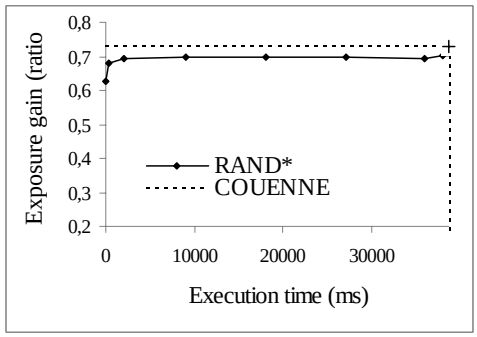
[31] Yao, D., Frikken, F. B., Atallah, M. J., Tamassia, R.: Private information: To reveal or not to reveal, *ACM Transactions on Information and System Security (TISSEC)*, **12**(1), 2008.



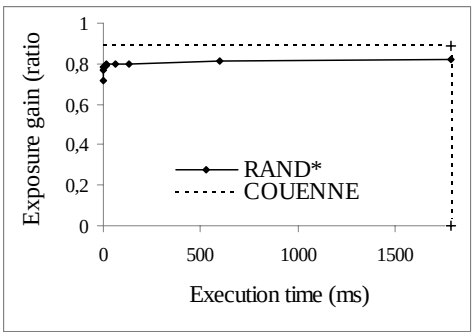
(a) Full Graph - ENRON



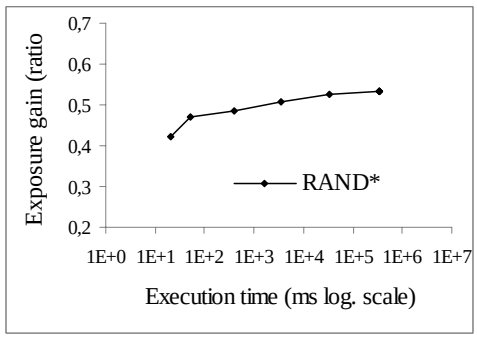
(b) Local Graphs - ENRON



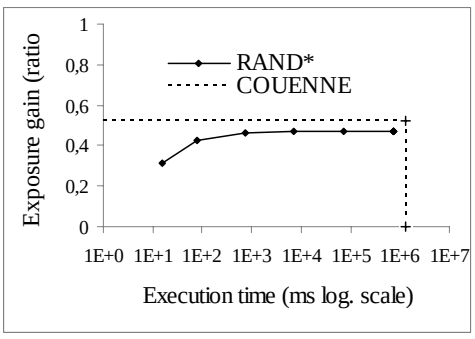
(c) Full Graph - MEDICAL



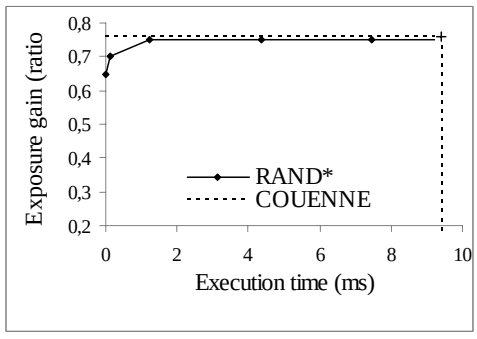
(d) Local Graphs - MEDICAL



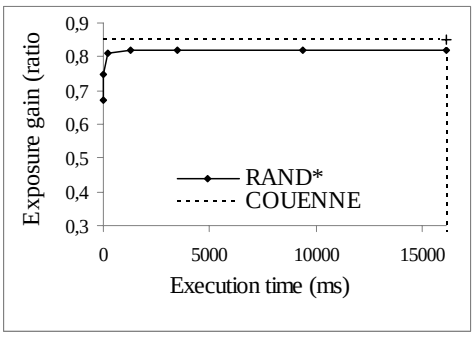
(e) Full Graph - SOCIAL



(f) Local Graphs - SOCIAL



(g) Largest Applications - ENRON



(h) Largest Application - MEDICAL

Figure 6. Exposure Gain Ratio