http://dx.doi.org/10.1287/ics.2015.0019

14th INFORMS Computing Society Conference
Richmond, Virginia, January 11–13, 2015
pp. 245–256

Cooperative Data Analysis in Supply Chains Using Selective Information Disclosure

Jörg Lässig
University of Applied Sciences Zittau/Görlitz, Department of Computer Science, Germany,
jlaessig@hszg.de

Michael Hahsler
Southern Methodist University, Bobby B. Lyle School of Engineering, Dallas, Texas, USA,
mhahsler@lyle.smu.edu

Abstract Many modern products (e. g., consumer electronics) consist of hundreds of complex parts sourced from a large number of suppliers. In such a setting, finding the source of certain properties, e. g., the source of defects in the final product, becomes increasingly difficult. Data analysis methods can be used on information shared in modern supply chains. However, some information may be confidential since they touch proprietary production processes or trade secrets. Important principles of confidentiality are data minimization and that each participant has control over how much information is communicated with others, both of which makes data analysis more difficult.

In this work, we investigate the effectiveness of strategies for selective information disclosure in order to perform cooperative data analysis in a supply chain. The goal is to minimize information exchange by only exchanging information which is needed for the analysis tasks at hand. The work is motivated by the growing demand for cross company data analysis, while simultaneously addressing confidentiality concerns. As an example, we apply a newly developed protocol with association mining techniques in an empirical simulation study to compare its effectiveness with complete information disclosure. The results show that the predictive performance is comparable while the amount of exchanged information is reduced significantly.

Keywords Data mining; information sharing; supply chain; privacy

1. Introduction

Over recent decades, our capability to collect, store and analyze data has significantly increased. Also, driven by recent discussions and developments, concerns regarding confidentiality, privacy issues and misuse of the collected data are more prevalent than ever. Enterprises are increasingly concerned about industrial espionage by potential competitors. The research area of privacy preserving data mining [3] tries to address these issue by developing specialized algorithms. These algorithms ensure that the privacy of sensitive data is preserved to a certain extend during data analysis. Most of these algorithms are developed with the aim of protecting private information of individuals (income level, age, etc.) while still enabling data analysis. In this paper, we deal with companies participating in a supply chain [5]. These companies have very different information protection goals than individuals and have an incentive to share certain information (e.g., logistics and demand forecast information) with partners in the supply chain. However, we argue that even in this situation some companies may prefer not to share all information (e. g., details about a proprietary production process or the change of a third party supplier) or it might be impractical to

share all information because of the shear volume (e. g., on what set of machines each items was produced). Therefore, the type of information to be shared and the incentive structure in this setting has some characteristics which are different from the standard problems addressed by privacy preserving data mining. In this paper, we will focus on the problem of identifying correlated features in vertically partitioned data sets and analyze how information can be selectively shared to achieve this goal in a cooperative fashion. We will discuss considerations about the trade-off between limited information disclosure and the ability to infer information about the root cause of certain problems based on examples in a supply chain context. Furthermore, we will touch very briefly upon how to avoid problems due to misaligned incentives.

The remainder of this paper is organized as follows: Section 2 reviews related work including privacy in data mining and data analysis in supply chains. A formal problem description of the scenarios investigated in this work is given in Section 3. Our new approach and suggested selective information disclosure setups are introduced in Section 4. Results of an empirical study of the approach is presented in Section 5. Section 6 describes considerations about how to implement the approach in a supply chain and how issues due to incentive misalignment can be addressed. Finally, Section 7 concludes the paper and addresses directions for future work.

2. Related Work

Related work to the topic must be reviewed in two directions. On one hand, the development of methods in the context of data mining and privacy must be discussed (Section 2.1), while, on the other hand, data analysis in the special application field—supply chains—as a field with a vivid history, has to be reviewed (Section 2.2).

2.1. Data Analysis and Privacy

After early ideas [8], seminal work introducing the idea to include privacy concerns in data mining methods has been done around the turn of the millennium by Agrawal and Srikant [3] and Lindell and Pinkas [19]. The former paper addresses the question of how to learn predictive models without accessing the exact information in individual data records. For example, decision trees can be learned from training data which is made imprecise using small changes of the original values. The latter paper introduces a model involving two parties which own confidential databases and need to perform a data mining task (decision tree learning with ID3) on the combination of both databases, without exposing information unnecessarily to the other party.

Early approaches of preserving privacy include microaggregation and k-anonymity. Microaggregation [6] is a technique to control disclosure for statistical databases. Individual records are aggregated into groups and then only these aggregated values are disclosed. To guard individual information from exposure, aggregates need to be constructed from at least k data points, where k is a constant controlling the level of protection. Closely related to microaggrigation is the concept of k-anonymity [7]. k-anonymity uses attribute suppression and generalization to ensures that each individual's information in the dataset cannot be distinguished from at least $k-1$ others and thus mitigating the tension between data utility and respondent privacy.

The two mentioned approaches try to protect the privacy of individuals in data sets published for statistical purposes. Work which comes closer to the scenario discussed in this paper is privacy preserving data mining. Here several parties need to analyze their data sets together without revealing confidential information unnecessarily to other parties. Various privacy preserving algorithms for classification, data clustering and association mining have been proposed during the last decade [28, 20, 1]. This covers methods to analyze vertically or horizontally partitioned data sets. Recently, alternatives based on standard data mining

methods instead of privacy preserving data mining methods have also been suggested. These include methods based on local cluster identifiers to obfuscate information [25].

In this paper we focus on a classification task for vertically partitioned data sets in a supply chain. However, in contrast to the above mentioned methods, which try to learn a global model from partitioned data, we are interested in learning local models, one for each involved participant in the supply chain, in a distributed manner. This not only reduces the amount of data to be shared, but these local models can be directly used by each participant to improve their processes and thus enhance the outcome of the supply chain as a whole.

2.2. Data Analysis in Supply Chains

Looking at the application domain for our analysis, there is a long history of research. In the 1900s, the work by Lee and Whang [15] discusses the benefits and connected problems of information sharing in the supply chain. Since then, the topic has not lost any of its importance. The huge variety and types of shared information in the supply chain is discussed in [17], considering information sharing as a basic building block for implementing tight coordination within the chain. Information typically shared includes information about inventory, order status, sales, demand forecasts, and the production schedules. The value of information sharing in supply chains has been broadly investigated [16, 13].

Li [18] finds that information sharing in a supply chain can be discouraged by certain direct and indirect effects of competition. On the other hand, improved profits and social benefits can facilitate information sharing. Further investigations of the barriers and performance of e-Integration in supply chains, with emphasize on the benefits, can be found in [10, 9]. Subramani [26] and Yu *et al.* [29] discuss the benefits for the participants of supply chains. The survey of Huang *et al.* [14] reviews over 100 publications which discuss the impacts of sharing production information on the supply chain dynamics.

Zhou and Brenton [31] consider the practical application of information sharing by investigating over 125 manufacturing companies in North America empirically. The study revealed that effective information sharing significantly enhances supply chain practice and that effective information sharing and effective supply chain practice are critical for the supply chain performance. In [12], software solutions for practical information sharing in supply chains in the form of logistics information systems are discussed and compared. In particular, hardware solutions like RFID are discussed in [4, 24].

Information sharing can be problematic in the face of horizontal competition between supply chain participants [18]. For example, a participant might have an incentive (e.g., financial or loss of reputation) to hide the fact that their component causes a certain problem. The approach in this paper addresses this issue by introducing selective information disclosure to protect information confidentiality and at the same time provides a mechanism that reduces the possibility to hide problems.

3. Formal Problem Description

A supply chain can be formalized as a directed acyclic graph $G = (V, E)$. The set of nodes V symbolizes the participants of the supply chain and the set of edges E the material and information flow. For simplicity, we consider here the scenario with a single final product produced by a single manufacturer represented by the sink node $s \in V$ with outdegree zero, and that each supplier $v \in V \setminus s$ supplies a single intermediate product which will be part of the final product. A very simple supply chain with one manufacturer s and 12 suppliers $v_1, v_2, \ldots v_{12}$ is shown in Figure 1. In this example, v_1 through v_{10} directly supply the manufacturer s, and v_{11} and v_{12} only indirectly supply s via v_2.

We assume that each final product is identified using a unique identifier $k \in K$ and that all intermediate products, which are part of the final product, can also be identified with the same identifier. The mapping between real product identifiers from different vendors is trivial

FIGURE 1. Example of a generic supply chain as a graph.

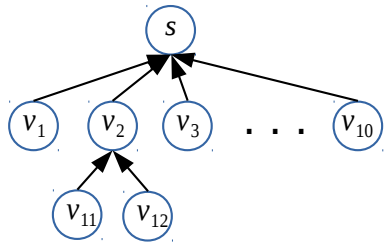


TABLE 1. Vertically partitioned data set $\mathcal{T}$ with class column c.

	T_{v_1}				T_{v_2}			...	$T_{v_{12}}$			
k	$t_{1,1}$	$t_{1,2}$	...	$t_{1,m(v_1)}$	$t_{2,1}$	...	$t_{2,m(v_2)}$	...	$t_{12,1}$	...	$t_{12,m(v_{12})}$	c
1	m 1	s 3	...	m 2	s 4	...	e 4	...	e 3	...	m 6	pass
2	m 3	s 9	...	m 2	s 1	...	e 2	...	e 7	...	m 1	fail
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\|K\|$	m 1	s 4	...	m 1	s 3	...	e 2	...	e 3	...	m 4	pass

Note. m ... machine, s ... supplier, e ... employee.

and standard practice in supply chains. Each node $v \in V$ also maintains an information table T_v with $m(v)$ columns and $|K|$ rows. Each column denoted by $t_{v,1}, t_{v,2}, \ldots, t_{v,m(v)}$ describes a single feature of the product, e. g., the source of a component or the machine on which a certain production step was performed. Each row represents one produced item and is associated with a single product identifier $k \in K$.

Each party, respectively each node in the graph, can assign certain class labels to each of its produced items. These class labels can be assigned to describe certain properties of the product. Examples are the results of system tests (pass/fail) or if a customer returned the product as defective. Here we assume that only the manufacturer assigns a class vector c to indicate if the product is defective.

The information about a final product is spread over all suppliers and therefore can be thought of as being stored as a vertically partitioned distributed data set. The distributed data sets can be joined together into the integrated virtual data set $\mathcal{T}$ by using the common key data K for matching. The resulting data set

$$\mathcal{T} = T_{v_1} \bowtie T_{v_2} \bowtie \ldots \bowtie T_{v_{|V|}} \bowtie c$$

represents the natural join of all columns (features) from all $|V|$ suppliers and the class vector c by matching the common key. Looking at this joint data set, all available data concerning all products can be considered as one table, with the product items as lines and the data of each node as a certain number of columns. Table 1 shows an example data set for the supply chain in Figure 1 with one manufacturer s and 12 suppliers v_1 through v_{12}. It contains data for $|K|$ products and one classification vector c. Partitions are shown using double lines.

Dependencies between c in a node v to the data T_u with $u \in V \setminus v$ may exist, if and only if, a path $p(u, v)$ in G exists. Let us call the node u with a path $p(u, v)$ in G an *influential node* of v and denote the set of influential nodes for v by $I(v)$, called *influential set*. For example, in Figure 1 the influential set of v_2 is $I(v_2) = \{v_{11}, v_{12}\}$. This motivates the idea, that class labels c_v could be characterized by using all the data T_u of nodes $u \in I(v)$. Assuming that the integrated data set $\mathcal{T}$ is available, this is a simple task. However, if parties of the supply chain are not interested in providing access to their internal data, then this analysis is not possible. In order to perform analysis without the need to collect all information in a central place, we developed the following protocol for selective information disclosure.

4. Protocol for Optimized Information Disclosure

The main idea of our approach is to reverse the roles that the participants in the supply chain typically play in analysis. Often the manufacturer s collects all information via the supply chain and performs the analysis. Here we are interested in classification, and we will show that it is possible to leave the detailed data with their owners (the suppliers) and instead supply them with the class information. Now the analysis can be conducted in a distributed manner by building and analyzing local models for each supplier, solving the confidentiality problem for them. Next, we will introduce several different cases which we call trivial case, direct case and remote case.

4.1. Trivial Case

In the trivial case, only a single party is involved ($v = u$) and all information is directly available. Any available method can be used to discover the dependencies between the columns in T_v and c. We do not discuss this case further.

4.2. Direct Case

In the direct case $u \in I(v)$ with $(u, v) \in E$. This restricts the analysis to only direct suppliers with a path $p(u, v)$ of length one. In order to discover the dependency at party u, party v discloses a set of key-value pairs $\Gamma = (k, c)$ from its classification to party u. Here the keys k establish a set of products, or product IDs, for which the classification c is known and of interest. Based on this information, party u selects the rows corresponding to the keys k in Γ and adds the class attribute c as an additional column. Different models $\mathcal{C}$ can be used to find dependencies. Examples are association rules, decision trees or statistical tests of dependency. Below we describe different scenarios. Without loss of generality, we will assume that the class attribute in the scenarios represents information of production failures.

Scenario 1: In scenario 1, v knows that a component supplied by $u \in I(v)$ passes quality control, but still a fraction of the components produce problems in the final product. v provides classification data in the form of $\Gamma = (k, c)$ tuples to u. Party u infers a model $\mathcal{C}$ to screen its products. If u identifies a problematic feature (e. g., a certain supplier for a sub-component) then u can decide to only deliver items which most likely will not cause faulty products to party v. This will decrease v's failure rate and the associated cost. Party u can also use the model to analyze and improve its production process (e. g., replace a components supplier or readjust a machine).

Scenario 2: In scenario 2, v does not know what component from what supplier causes the problem. Here v provides tuples $\Gamma = (k, c)$ to all $u \in I(v)$. Each supplier locally creates a model $\mathcal{C}$. Then each supplier analyzes the model to see if its production process is associated with the classification. If one supplier finds an association, then he can report this to v and resolve the problem in the same way as in scenario 1.

Scenario 3: Scenario 3 is similar to scenario 2, but the problem is caused by the interaction of several (often two) features from different unknown suppliers. The tuples are again provided to all suppliers and each supplier checks for (potentially weak) associations. Since both notify v, v can work with both suppliers to ensure that the components are matched to avoid the interaction.

4.3. Remote Case

This case starts exactly the same as the direct case, but party v which received the tuples Γ finds out that a feature which represents one of its own suppliers w is associated with the class attribute c in Γ. In this situation, party u can ask party v, if the classification information can be propagated further down the supply chain to its supplier w. The procedure at w is exactly the same as in the direct case.

Table 2. Parameters used for the simulation study.

Parameter	Symbol	Value
Number of suppliers	$\|V\|$	10
Features per supplier	$m(v)$	$[5, 20]$
Values per feature	range$(t_{v,i})$	$[2, 10]$
Base defect rate	e_{base}	1%
Feature defect rate	e_{feature}	100% or 5%
Number of shared tuples	$\|\Gamma\|$	1,000 or 10,000
Number of simulation runs	n	100

5. Simulation Study

In this paper, we conduct a simulation study to investigate the feasibility of selective information disclosure. Both, the trivial case and the remote case can be seen as either a simple special case or a recursive application of the direct case, respectively. Therefore, we concentrate on the three scenarios of the direct case in this study. The aim is to compare the effectiveness of the selective information sharing approach with complete information disclosure, i. e., all suppliers have to share all information with the manufacturer.

5.1. Modeling the Data

For the simulation study, we simulate a single manufacturer with $|V|$ suppliers. Table 2 contains the parameters used in the simulation study. For each supplier v we randomly choose the number of features $m(v_i)$ of the supplied component from the integer range $[5, 20]$. Each feature can describe a distinct production step or a subcomponent. For each feature t we randomly choose the number of different possible values from $[2, 10]$. These values might indicate, for example, the person performing a production step or the particular supplier of a sub component. Each manufactured product has a chance of being defect governed by the base defect rate of 1%. For the effected feature/value combination the defect rate is increased to either 100% or 5% to signify that all products produced this way are defect or the defect rate just increases from 1% to 5%. For the number of tuples shared, we consider 1,000 and 10,000 randomly chosen tuples. Tuples can also be selected in a different way to balance the class labels or create any class distribution such that the supplier cannot gain information about the actual distribution (i. e., what proportion of products by the manufacturer are effected).

5.2. Method for Finding Significant Dependencies

Dependencies can be found by many different data mining or statistical methods (see, e. g. [27]). In this study we focus on discrete features. For example, the same production step can be done with three different machines resulting in a nominal feature with three different values. Here we will use association rules to relate feature values with the class attribute using the regular support/confidence framework [2]. However, to detect significant association rules we will employ a one-sided Fisher's exact test for the analysis of 2×2 contingency tables which was described for association rules in [11]. This test is similar to the χ^2-test, but is also appropriate for small sample sizes. We accept all associations with a p-value less than a predefined significance level α as found causes of the defect. In the following we will discuss how we correct for multiple comparisons and the impact of the number of shared tuples $|\Gamma|$.

Correction for multiple comparisons Association rule mining is known to produce a large number of rules which have to be tested. Performing many statistical test while focusing on the strongest results of all tests is known to produce an increased false positive rate. To maintain the desired significance level, we have to correct for multiple comparisons. We will use Bonferroni correction [22] which divides the overall (familywise) significance

level α^* by the number of tests performed, i. e., $\alpha = \alpha^*/m$, where m is the number of tests. Note that the number of tests is not the number of mined association rules and thus the tests actually performed, but the number of all possible association rules. For the application in this paper, it is sufficient to mine rules with a single antecedent (left-hand-side) item and the item that indicated the important value of the class attribute (e. g., that the product is affected by the problem) in the consequent (right-hand-side). Therefore, the number of possible associations is the number of feature values and we divide α by that number. For the simulations in this paper we use an overall significance level of $\alpha^* = 5\%$.

Note that for correction there is a difference between complete disclosure, where the analyst has access to all the data, and selective disclosure, where each company has only access to its own data and some class information. In the complete disclosure case, the number of tests performed is equal to the total number of feature values used by all suppliers. For the selective disclosure case, each company only knows about the number of feature values it uses and thus only performs a number of tests equal to its own feature values. This results in two issues for selective disclosure: (1) the effective comparison-wise significance level is corrected less, and (2) the used significance level potentially varies between companies, depending on the number of feature values each uses. We will investigate this influence in the simulation results.

Number of shared tuples $|\mathbf{\Gamma}|$ The number of shared tuples represents the information that the manufacturer exposes to other participants in the supply chain and reducing the number of tuples exposes less information of the manufacturer. Since the tuples represent the products which can be used for analysis, they define a sample of transactions for association rule mining. Several authors have worked on establishing bounds for sampling strategies for association rules. Mannila *et al.* [21] suggests the use of Chernov bounds on the number of transactions containing a itemset in a sample. Zaki *et al.* [30] built upon the theoretic work in [21] and show that there is a relationship between the support $\tau = \text{supp}(X)$ of itemset X and the needed sample size n if we accept a relative error for the measured support of ε at a given confidence level $1 - c$:

$$n = \frac{-2\ln(c)}{\tau\varepsilon^2} \,. \tag{1}$$

The sample size depends on each itemset's support, however, the authors suggest to set τ to the minimum support threshold used for mining associations. In this case the sample size is large enough that the error rate ε holds at the given confidence level even for the least frequent itemsets found and we get better estimates for more frequent itemsets.

For a minimum support of 1%, an accuracy level of 95% and a confidence level of 95%, Equation (1) gives a sample size of approximately 240,000. However, the Chernov bound is very loose and evaluation of sampling in practice showed that a size much smaller than the one obtained by Equation (1) is typically sufficient [30]. We experimented with different sample sizes (number of tuples $|\Gamma|$) in our simulation and choose to report results for 10,000 for a large sample size and 1,000 for a small sample size.

5.3. Results

We present the results for Scenarios 2 and 3 of the direct case and compare selective information disclosure with complete information disclosure. We omit the results for Scenario 1 because this scenario is a simple special case of Scenario 2 with only a single supplier. All simulations use the common parameters shown in Table 2. Figure 2 shows the distribution of the number of different feature values in the simulation with an average of around 744 feature values.

In Scenarios 2 and 3, the manufacturer does not know what supplier is responsible for the defect. For complete disclosure, all suppliers need to share their information about a set of products defined by the manufacturer. For selective information disclosure, the manufacturer

FIGURE 2. Distribution of the number of simulated feature values for the 100 simulation runs.

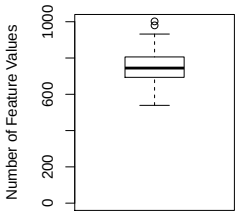
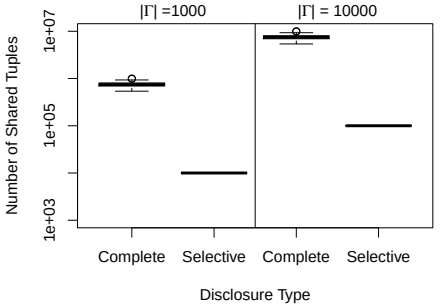


FIGURE 3. Amount of shared information for complete and selective information disclosure (logarithmic scale).



shares the set of $\Gamma = (k, c)$ tuples with all its direct suppliers. For comparison, we assume that the products in Γ are also the products that the manufacturer collects information about from the suppliers in the complete disclosure case. Figure 3 shows that while the information exchange for selective disclosure is fixed at $|\Gamma| \times |V|$ (number of tuples times the number of suppliers), the information exchanged for complete disclosure is several orders of magnitude larger and results in the manufacturer having access to all the potentially sensitive supplier data.

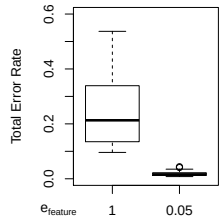
Scenario 2 In this scenario, a single feature value causes the defect. We find all association rules with two items and then calculate for each the p-value for Fisher's exact test. We correct the significance level for the number of tests and report the antecedent (left-hand-side) of significant rules as found feature values. Figure 4 show the results for 100 simulation runs. The total defect rate is increased from 1% (the base defect rate) to on average 25.4% or 1.7% when the introduced error for the effected feature value e_{feature} is 100% or 5%, respectively.

Figure 4(b) shows that the approach is able to find the problem feature value (true positive rate) if it always results in a defective product ($e_{\text{feature}} = 1$), or if the number of tuple is large ($|\Gamma| = 10,000$). For a smaller number of tuples and a smaller introduced error, even with complete disclosure only in 30% of the cases, we can identify the feature value causing the defect. Interestingly, for selective disclosure the chance of identification increases to 34%. The reason is that, in the selective disclosure, the correction for multiple comparisons is weaker, resulting in more positives. There is very little difference between the two disclosure forms in terms of ability to detect the source of the defect. On the other hand, Figure 4(c) shows that the rate of false positives, i. e., feature values incorrectly identified as reasons for the problem also increases for selective disclosure from about 0.02% to 0.35%. This number reflects the chance that a feature value is incorrectly identified as the source of the problem. It is much higher for selective disclosure since each supplier only corrects for its own multiple comparisons.

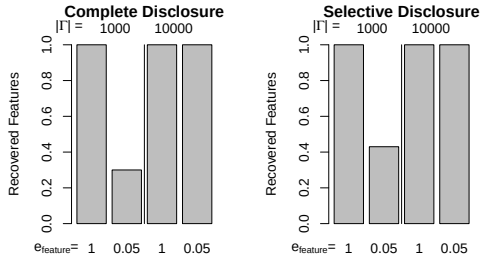
Scenario 3 In this scenario the interaction between two features from different suppliers causes the problem. Figure 5(a) shows the total error rate increases from 1% to 5% for $e_{\text{feature}} = 1$ and from 1% to 1.2% for $e_{\text{feature}} = .05$. The defect rate is significantly smaller than in Scenario 2 since a defect only occurs if the two affected feature values appear together in a product.

Figure 5(b) shows that for $e_{\text{feature}} = 1$, both selective and complete disclosure detect almost always the problematic feature values, but for $e_{\text{feature}} = .05$ detection of the source of the defect is very difficult with less than 1% of detections for a small tuple size ($|\Gamma| = 1,000$) and around 14% for a large tuple size ($|\Gamma| = 10,000$). Selective disclosure again provides

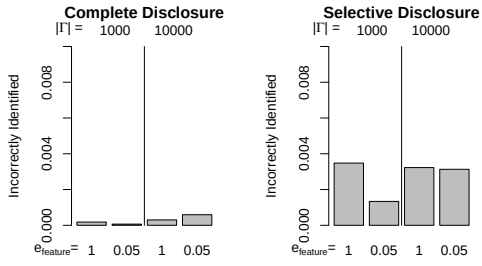
FIGURE 4. Simulation results for Scenario 2 with a single problematic feature. Total defect rates are shown in (a). Comparison of (b) correctly detected feature values and (c) incorrectly identified feature values between complete disclosure and selective disclosure.



(a)

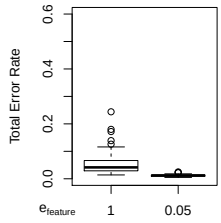


(b)

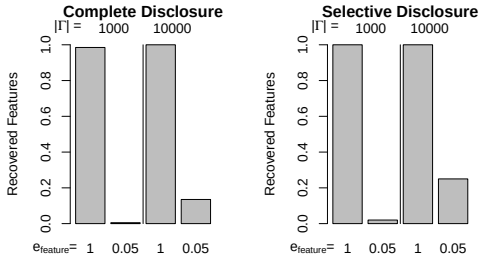


(c)

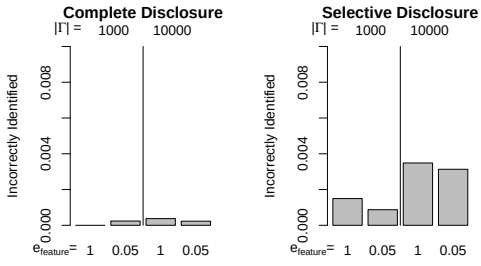
FIGURE 5. Simulation results for Scenario 3 with two interacting features. The total defect rate is shown in (a). Comparison of (b) correctly detected feature values and (c) incorrectly identified feature values between complete disclosure and selective disclosure.



(a)



(b)



(c)

similar results with slightly higher detection rates due to the weaker correction for multiple comparisons. Also, the rate of incorrectly identifying feature values is higher for selective disclosure and reaches 0.35%, as shown in Figure 5(c).

6. Implementation Considerations

The free software package R for statistical computing has been used[1] for the proof of concept of the proposed method in this publication. To make the approach usable in a practical

[1] `http://www.r-project.org/`, last accessed September 12, 2014

setting where several companies with different analytical capabilities and software environments are involved, we need to support information sharing and the analysis process. We plan to implement the approach based on the plug-in mechanism in the open source data mining framework RapidMiner[2] which can be easily deployed to the supply chain participants. The participants make their information available to the locally running RapidMiner instance, but are guaranteed that the information will only be used locally and not shared. The locally running RapidMiner instances in each company will be securely connected to a central coordination server where supply chains with an arbitrary numbers of participants can be modeled and configured based on graph structures as shown in Figure 1. This coordination server can be located at the manufacturer or at a trusted third party since it just manages the communication, but never accesses any confidential data. Information exchange (class information and an indication if a participant's features are possibly associated) only takes place directly between participants and only after the participants' consent. An example would be that the manufacturer provides some class labels indicating a rare hardware failure and would like to address this problem with several suppliers. At this point the manufacturer consents to sharing the class information and articulates to the suppliers the reason for the requested analysis. The suppliers need to manually consent to the analysis. After their consent, the class information is securely transferred directly from the manufacturer and the analysis is automatically performed. Detailed analysis results are available to each supplier and an indication if significant associations were found is securely transmitted to the manufacturer. Note, that once a party consents, the analysis is fully automatic and a party cannot hide the results from the manufacturer. This is an important feature of the process because it helps to mitigate the issue with the alignment of incentives [23] when the analysis is carried out locally by the suppliers. It removes the option for not admitting that a component is responsible for the problem and therefore reduces adverse effects on the whole supply chain (e.g., reduced sales due to delays in fixing an issue).

7. Conclusion

In this paper we have explored how to support collaborative data analysis in a supply chain while protecting confidential data through a mechanism of selective information exchange. Instead of privacy preserving data mining algorithms, standard non-privacy preserving approaches can be used with the proposed protocol. After introducing the protocol we discussed several scenarios and conducted a simulation study to show that selective information disclosure can produce results comparable to complete disclosure. This motivates further investigation of the approach. Especially interesting is to explore how other classification models (e. g., decision trees) perform in this setting. Also, the amount of data being disclosed can be potentially further reduced by employing strategies from progressive sampling where we start with very little information and successively increase the amount disclosed till the quality of the resulting models is sufficient.

It is planned to implement a supply chain package based on the plug-in mechanism of the open source software RapidMiner to perform a case study in a real setting.

Acknowledgment

Part of the work has been done during a research visit of J. L. at the Southern Methodist University in Dallas, Texas. The author would like to thank the Bobby B. Lyle School of Engineering and M. H. for the invitation and support.

[2] `http://www.rapidminer.com/`, last accessed September 12, 2014

References

[1] Charu C Aggarwal and Philip S Yu. A general survey of privacy-preserving data mining models and algorithms. In *Privacy-Preserving Data Mining. Advances in Database Systems*, Volume 34, pp 11–52, Springer, 2008.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington D.C., May 1993.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.

[4] Zaheeruddin Asif and Munir Mandviwalla. Integrating the supply chain with RFID: A technical and business analysis. *Communications of the Association for Information Systems*, 15, 2005.

[5] David Blanchard. *Supply Chain Management Best Practices*. John Wiley & Sons, 2nd edition, 2010.

[6] Josep Domingo-Ferrer and Josep Maria Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189–201, 2002.

[7] Josep Domingo-Ferrer and Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[8] Vladimir Estivill-Castro and Ljiljana Brankovic. Data swapping: Balancing privacy against precision in mining for logic rules. In *Data warehousing and knowledge discovery*, pages 389–398. Springer, 1999.

[9] Barbara B Flynn, Baofeng Huo, and Xiande Zhao. The impact of supply chain integration on performance: a contingency and configuration approach. *Journal of Operations Management*, 28(1):58–71, 2010.

[10] Markham T Frohlich. e-Integration in the supply chain: Barriers and performance. *Decision Sciences*, 33(4):537–556, 2002.

[11] Michael Hahsler and Kurt Hornik. New probabilistic interest measures for association rules. *Intelligent Data Analysis*, 11(5):437–455, 2007.

[12] Petri Helo and Bulcsu Szekely. Logistics information systems: an analysis of software solutions for supply chain co-ordination. *Industrial Management & Data Systems*, 105(1):5–18, 2005.

[13] Craig A Hill and Gary D Scudder. The use of electronic data interchange for supply chain coordination in the food industry. *Journal of Operations Management*, 20(4):375–387, 2002.

[14] George Q Huang, Jason SK Lau, and KL Mak. The impacts of sharing production information on supply chain dynamics: a review of the literature. *International Journal of Production Research*, 41(7):1483–1517, 2003.

[15] Hau L Lee and V Padmanabhan. Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558, 1997.

[16] Hau L. Lee, Kut C. So, and Christopher S. Tang. The value of information sharing in a two-level supply chain. *Management Science*, 46(5):626–643, 2000.

[17] Hau L Lee and Seungjin Whang. Information sharing in a supply chain. *International Journal of Manufacturing Technology and Management*, 1(1):79–93, 2000.

[18] Lode Li. Information sharing in a supply chain with horizontal competition. *Management Science*, 48(9):1196–1212, 2002.

[19] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. In *Advances in Cryptology–CRYPTO 2000*, pages 36–54. Springer, 2000.

[20] Olvi L Mangasarian and Edward W Wild. Privacy-preserving classification of horizontally partitioned data via random kernels. In *Proceedings of the 4th International Conference on Data Mining*, pages 473–479, 2008.

[21] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *AAAI Workshop on Knowledge Discovery in Databases (KDD-94)*, pages 181–192, Seattle, Washington, 1994. AAAI Press.

[22] Rupert G Miller. *Simultaneous Statistical Inference*. Springer-Verlag, 2nd edition, 1981.

[23] V. G. Narayanan and Ananth Raman. Aligning Incentives in Supply Chains. *Harvard Business Review*, 82(11):94–102, November 2004.

[24] Nico Schlitter, Florian Kähne, Stiefen T Schilz, and Holger Mattke. Potential and problems of RFID-based cooperation in supply chains. In *Proceedings of Hamburg International Conference of Logistics (HICL2007)*, pages 147–164, 2007.

[25] Nico Schlitter and Jörg Lässig. Distributed privacy preserving classification based on local cluster identifiers. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1265–1272, IEEE, 2012.

[26] Mani Subramani. How do suppliers benefit from information technology use in supply chain relationships? *MIS Quarterly*, 28(1):45–73, 2004.

[27] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson Education, 2006.

[28] Hwanjo Yu, Jaideep Vaidya, and Xiaoqian Jiang. Privacy-preserving svm classification on vertically partitioned data. In *Advances in Knowledge Discovery and Data Mining*, pages 647–656. Springer, 2006.

[29] Zhenxin Yu, Hong Yan, and TC Edwin Cheng. Benefits of information sharing with supply chain partnerships. *Industrial Management & Data Systems*, 101(3):114–121, 2001.

[30] Mohammed J. Zaki, Srinivasan Parthasarathy, Wei Li, and Mitsunori Ogihara. Evaluation of sampling for data mining of association rules. In *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications*, pages 42–50. IEEE Computer Society, 1997.

[31] Honggeng Zhou and WC Benton Jr. Supply chain practice and information sharing. *Journal of Operations Management*, 25(6):1348–1365, 2007.