Towards Cooperative Predictive Data Mining in Competitive Environments

Viliam Lisý, Michal Jakob, Petr Benda, Štěpán Urban, and Michal Pěchouček

Agent Technology Center, Dept. of Cybernetics, FEE, Czech Technical University
Technická 2, 16627 Prague 6, Czech Republic
{lisy, jakob, benda, urban, pechoucek}@agents.felk.cvut.cz

Abstract. We study the problem of predictive data mining in the competitive multi-agent setting, in which each agent is assumed to have some partial knowledge needed for correctly classifying a set of unlabelled examples. The agents are self-interested and therefore need to reason about the trade-offs between increasing their classification accuracy by collaborating with other agents and disclosing their private classification knowledge to other agents through such collaboration. We analyze the problem and propose a set of components which can enable cooperation in this otherwise competitive problem. These components include measures for quantifying private knowledge disclosure, data-mining models suitable for multi-agent predictive data mining, and a set of strategies by which agents can improve their classification accuracy through collaboration. The overall framework and its individual components are validated on a synthetic experimental domain.

1 Introduction

We study the case of multiple self-interested parties (termed *agents* further on) working on a common but partitioned predictive data mining task[1] in a competitive environment. The knowledge used in solving the data mining task is considered a valuable asset of each party because accurately predicting data classifications is assumed to give the party a competitive advantage. One of the ways the classification accuracy can be improved is by exchanging knowledge with other parties. Given the competitive nature of the domain, such exchange cannot be done on an arbitrary basis but needs to follow sound strategies that consider knowledge lost and gained during each transaction. The ability to implement such strategies is thus a necessary precondition to enabling cooperation between the agents.

In this paper, we outline a framework and describe a set of specific methods and techniques that make such an implementation possible. First, we develop a set of private knowledge disclosure metrics that measure the classification knowledge lost in data and model exchanges between the agents. We identify

[1] Different agents have different datasets but all the datasets are sampled from a single underlying model

a set of operations that need to be supported by the underlying classification models to make them applicable in the cooperative setting, in particular the possibility to effectively evaluate private knowledge metrics and to incrementally accommodate prediction knowledge obtained from other agents and, vice versa, to effectively extract knowledge to be shared. Not all classification models fulfil these requirements – we identify the Naive Bayes classifier as a particularly suitable class of models for the semi-cooperative prediction, and we show how individual operations can effectively be implemented for this class. We then describe several cooperative strategies that the agents can use to improve their classification capability. For each strategy, we measure the resulting improvement of agent's classification capability and set it against the loss of privacy entailed.

The work presented should be viewed as an initial step towards enabling cooperation in predictive data mining in a community of self-interested and competitive agents.

2 Related Work

Over the last decade, privacy preserving data mining (PPDM) [1] has attracted considerable attention. One of the main objectives of the field is designing data transformations which allow publishing data without losing privacy. In most cases, the emphasis is on protecting the privacy of *individuals* described by the data records – not on the protection of knowledge contained in the data set as a whole. Consequently, losing exact information about a small number of records is considered unacceptable while complete models learned from the data can be freely shared unless they allow the derivation of individual records [2].

This is well acceptable for the preservation of private information about individuals, such as medical records, but it is not sufficient in reasoning about the preservation of private data, knowledge, and know-how of companies. Almost any data mining result, a cluster model, a frequent sequence, or an association rule derived from the private company data can have commercial potential and should not be blithely shared without further assessment.

The subfield of PPDM that deals with this kind of private information, i.e. the knowledge contained in collections of data rather than individual records, is termed *corporate privacy* in [3] or *knowledge hiding* e.g. in [4]. There is, however, no general framework to reason about private knowledge disclosure – each technique has its own methods and measures of quality of private knowledge hiding. The research closest to the focus of this paper is classification rules [5] and association rules [6] hiding. The objective of these methods is to transform the data by resampling, reduction and/or perturbation so that a specified set of rules cannot be mined from the data; the outcome is measured in terms of the number of rules successfully hidden, the number of original data items modified and the number of new rules introduced by the transformation.

In contrast to the above, in this paper, we do not aim to conceal any specific classification rules; instead, we aim to measure and minimize the overall knowledge disclosed about private classification models possessed by individual agents

(and possibly obtained by generalizing their private data). As far as measures for quantifying privacy loss in PPDM are concerned, a summary is presented in [7]. Existing measures, often utilizing the notion of information entropy, are designed to measure the disclosure of a specific kind of knowledge. Moreover, except for a few exceptions (e.g. [8] for association rules), existing measures deal with individual privacy and are therefore not applicable to our problem.

On the other hand, privacy loss measured in terms of changes to information entropy has been used in multi-agent system research outside the data mining field (e.g. in multi-agent meeting scheduling [9], or multi-agent planning [10].

3 Problem Description

The problem addressed is *semi-cooperative classification* in competitive multi-agent domains. Each agent aims to accurately classify a set of unlabelled examples drawn from the same data distribution. In order to improve its classification accuracy, the agent can either exploit its own data and models or request additional knowledge from other agents. More specifically, the problem is defined as follows. There are multiple agents in the system, each containing its:

- set of labelled training data D_i
- set of unlabeled data for classification C_i (termed *task data*)
- a data mining algorithm M_i that can construct a classification model $M_i(S)$ from any labelled set of data S

The task of each agent is to classify its unlabelled task data C_i. In doing so the agent employs a particular *strategy* which involves exploiting its training data D_i and communicating with other agents about their models, data and classifications. Each strategy results in a specific classification accuracy (measured by a designated accuracy metric – see Section 4.1) and disclosure of a specific amount of private knowledge sourced externally (measured by a designated private knowledge loss metric – Section 4.2).

4 Relevant Measures

In this section, we introduce measures for evaluating agent's knowledge sharing actions with respect to (1) the loss of private knowledge and (2) the increase in classification accuracy. We briefly introduce the measures here and discuss their properties and suitability in a more detail in Section 7.

4.1 Private Knowledge Loss

According to our literature survey, no metrics for measuring the loss of information about a private classification model has been so far reported. In the following, we describe and analyze how symmetrised Kullback-Leibler divergence and mutual information can be used for this purpose. The application of these measures to quantify private knowledge loss is novel.

Both proposed measures are based on the following idea. If an agent sends some information originating from its private classification model (e.g. classified examples or partial models) to another agent, this information can be used to (approximately) reconstruct the private classification model of the sending agent. For example, if the agent discloses a set of feature vectors classified by the private model, these samples can be used to train a new model that approximates the original one. The proposed measures therefore evaluate the distance between the original private model and the (hypothetical) reconstructed model and use it as a quantification of the private classification knowledge loss.

The proposed measures assume that any classifier can be interpreted in two ways. Either as a representation of a joint probability distribution over the feature space and the classification labels, or as a realization of a function assigning a unique label to each feature vector. Although the first representation is more informative, it is often hard to extract from certain classes of classification models. For example, the k-nearest neighbour classifier approximates the distribution by the ratio of examples of individual classes in the neighbourhood of the feature vector; decision trees can approximate the distribution for a feature vector by the ratio of class frequencies of training examples corresponding to the leaf the feature vector belongs to. For the cases where the distribution can be extracted, we propose a measure based on the probabilistic similarity measure between the original distribution represented by the private classifier and the distribution realized by the reconstructed classifier.

On the other hand, extracting the distribution from a set of rules derived by ILP or similar method is not straightforward. Moreover, the internal structure of some classifiers is not always accessible to the agent (e.g. in the case of external libraries) and the classifier can be accessed only as a black-box producing a classification (class label) for any input feature vector. If classifications are the only available information about a classifier, we propose a metric based on information theory that measures the information present in the classifications produced by the reconstructed model about the classifications corresponding to the original private model.

Symmetrized Kullback-Leibler Divergence Kullback-Leibler (KL) divergence (also termed *information divergence* or *information gain*) is a standard measure used in probability theory to compute the similarity of two probability distributions. For two discrete probability distributions P and Q over the same random variable (the same feature space in our case), the KL divergence of Q from P is defined as

$$D_{kl}(P\|Q) = \sum_x P(x) log_2 \frac{P(x)}{Q(x)} \tag{1}$$

where x in the sum iterates over the range of the random variable. KL divergence in this form is not symmetric, i.e., $D_{kl}(P\|Q) \neq D_{kl}(Q\|P)$ in most cases. That is why a symmetrised form of Kullback-Leibler divergence is often used

$$D_{skl} = D_{kl}(P\|Q) + D_{kl}(Q\|P) \tag{2}$$

The iteration over the range of the random variable corresponds to iterating over the whole feature space in the case of a distribution realizing a classifier. This is generally computationally expensive for large distributions. For example in our case of 10 features with 10 possible values each, the size of the feature space is 10^{10}. However, for some specific classifiers, the value of KL-divergence can be approximated or even exactly computed in a significantly more effective way. (See Section 5.2).

Mutual Information Mutual information is a standard information-theoretic measure quantifying the mutual dependence of two random variables. It represents the amount of uncertainty about one random variable that is removed by knowing the value of the other random variable. This measure can be used to measure how much information a model reveals about a data set or about the classifications produced by a model regardless of the structure of the classifier.

For random variables X and Y, mutual information is defined as

$$I(X:Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log_2 \frac{P(x,y)}{P(x)P(y)} \tag{3}$$

Mutual information can be used to compute both the amount of information a model reveals about a data set and the amount of information one model contains about another model. We focus on the second case here. Classifications produced by the private model and by its approximation correspond to the two random variables in the formula. The only step needed is to estimate the probability distribution $P(X,Y)$, i.e., a normalized matrix specifying how frequently one classifier classifies a feature vector to class x while the other classifies the same vector to class y, which may or may not be the same. In the case of a smaller feature space, this probability distribution can be computed precisely by iterating over the whole feature space, similarly to the case of the distribution-based metric. Such explicit computation is not possible for large feature spaces. Instead, we therefore sample the two classification models on a chosen subset of the feature space and estimate the probability distribution $P(X,Y)$ from the confusion matrix resulting from the two sets of produced classifications.

A big advantage of this measure is its complete independence on the classifier implementation. A disadvantage of the mutual information metric is the amount of samples needed for good approximation of the distribution. We have performed a simple synthetic experiment that demonstrates this problem and shows some basic properties of the metric. We have simulated classification into 10 classes as in the experiments presented later. We were measuring the information that an imprecise classifier contains about an ideal classifier that always assigns the correct class. The probabilities in formula 3 can be estimated from the confusion matrix of the classifier, so we were generating only that matrix in

this experiment. Correct classification was provided with probability p, otherwise the classification was random to any of the remaining classes with uniform distribution. Figure 1 shows the dependence of the mutual information metric on the number of generated classifications in the confusion matrix in a single run of the experiment. The probability of correct classification were set to $p = 0$, $p = 0.5$ and $p = 1$, respectively, and the results shown are the mean of thousand runs. The variance was very high for less than 100 samples but decreased for more samples; the results obtained from more than 1000 samples were already almost identical across individual runs.

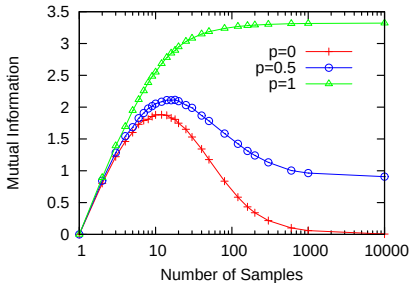

Fig. 1. Dependence of the mutual information metric on the number of samples used to construct the confusion matrix (from which the metric is calculated) for different probabilities of correct classification.

As we expected, when there is enough data samples, the mutual information between the correct classification and a classifier that always misclassifies into one of the nine wrong classes is almost zero. If the classifier classifies a half of the examples correctly, the mutual information is higher, and it is the highest when the classifier classifies all examples correctly. In the latter case of correct classification, it converges to the value of 3.32 bits, which corresponds to distribution $P(X, Y)$ with 10 classes and the probability uniformly distributed along the main diagonal.

The inaccurate probability estimations caused by a smaller number of samples lead, besides high variance, to a misleading increase in the measured information disclosure e.g. incorrectly indicating that ten random classifications reveal more information about the private model than a thousand classifications that are correct in half of the cases. In order to obtain reliable assessment of information disclosure using the mutual information metric, at least a thousand samples are needed.

4.2 Classification Accuracy

There are many measures for assessing classification accuracy. Some of them are based on ratios of true positive and true negative classifications or consider different costs of misclassification to different classes – see e.g. [11] for an overview.

The primary focus of this paper is on private knowledge preservation. We therefore use a ratio of correctly classified examples to all examples on a test data set as a basic classification accuracy measure.

5 Naïve Bayes for Multi-Agent Data Mining

A data mining model suitable for semi-cooperative multi-agent classification task should satisfy several criteria:

- *creating and joining sub-models* If agents exchange sub-models, their creation and joining should be possible and computationally efficient.
- *confidence* The model should be able to output the degree of confidence in the classification to allow the agents to decide when they need some extra information from other agents.
- *effective measures computation* The model should allow fast computation (or approximation) of relevant measures, in particular the private knowledge loss.
- *compact model size* The models should have a compact representation to allow sharing in case of limited communication bandwidth

One of the models that satisfies all the above properties is Naïve Bayes classifier. Joining two models is straightforward if they are represented as frequencies instead of probabilities (see Section 5.2). Creating a model requires only one iteration through the data set. The classifier output in the form of a posterior probability is suitable confidence measure and as we show below, the suggested privacy measures can be computed exactly in a reasonable time for this model.

5.1 Naïve Bayes Classifier

Let X is an n-dimensional feature space, X_i is a random variable representing the i-th component of the feature vector, and S is a set of all classes. Naive Bayes then classifies a given sample $(x_1, \ldots, x_n)$ into class

$$d = \arg\max_{s \in S} \left[P(S = s) \prod_{i=1}^{n} P(X_i = x_i | S = s) \right] \tag{4}$$

The product in the formula approximates the joint probability

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n | S = s) \tag{5}$$

under the assumption of the conditional independence of individual features.

5.2 Joining Naïve Bayes Models

The model sharing strategies require the ability to join classification models. In our framework, this is implemented in the following way. Naïve Bayes classifier is composed from a set of probability distributions

$$M = \bigcup_{s \in S} \{P(s), P_1(x_1|s), \ldots, P_n(x_n|s)\} \tag{6}$$

where $x_1 \in X_1, \ldots, x_n \in X_n$. We have implemented the model as

$$\bigcup_{s \in S} \{f(s), f_1(x_1|s), \ldots, f_n(x_n|s)\} \tag{7}$$

where $f(s)$ is the frequency of class s in the training set and $f_i(x_i|s)$ is the frequency of i-th attribute in the subset of the training data corresponding to class s. The number of data samples in the training set used to train the classifier can be directly determined from the implemented model as $a = \Sigma f(s)$. Probability $P(s)$ is then given by $\frac{f(s)}{a}$ and probabilities $P_i(x_i|s) = \frac{f_i(x_i|s)}{f(s)}$

Two models M^1 and M^2 are then joined as:

$$\bigcup_{s \in S} \{f^1(s) + f^2(s), f_1^1(x_1|s) + f_1^2(x_1|s), \ldots, f_n^1(x_n|s) + f_1^2(x_n|s)\} \tag{8}$$

KL-Divergence for Naïve Bayes The Naïve Bayes classifier and its feature independence assumption allows to significantly simplify the computation of certain measures. A simplified calculation of the otherwise computationally expensive KL-Divergence for two probabilistic distributions P_1 and P_2 representing two Naïve Bayes classifiers is shown below[2]

$$D_{kl}(P_1 \| P_2) = \sum_{s \in S} P_1(s) \left(log_2 \frac{P_1(s)}{P_2(s)} + \sum_{i=1}^{n} \sum_{x_i \in X_i} P_1(x_i|s) \, log_2 \frac{P_1(x_i|s)}{P_2(x_i|s)} \right) \tag{9}$$

where n is the number of features, X_i are the domains of the features and S is it set of classes.

6 Cooperation Strategies

As described in Section 3, the agents can use a cooperation strategy in order to improve their classification accuracy. The strategy can either involve requesting information about other agents' private models or asking the other agents to classify a set of unlabelled data using their private models. In the following description of the strategies, we use the term *requestor* for the agent aiming to improve its classification using the private knowledge of the other agents (which are termed *providers*).

6.1 Model and Sub-model Sharing

In this strategy, the requestor asks the providers for a partial or degraded version of their private models. In our experiments, the degraded version is a model trained only on a fraction of the dataset; other means of degradation, such as

[2] Detailed derivation is available on request from the authors.

adding noise to the model, or defining the model only on a subset of the feature-space, can be employed. The providers willing to share some information create partial models degraded to a level satisfying their private knowledge disclosure constraints and send them to the requestor. The requestor then merges the information provided in the received partial models into its own private model and uses the resulting improved model to classify its task data. In the case of the Naïve Bayes model, the merge algorithm has been described in Section 5.2.

6.2 Data Classification

In the case of the data classification strategy, the requestor first classifies its task data using its private classifier and sorts the results according to the confidence of the classification (the a posterior probability in the case of the Naive Bayes model). It then decides about the fraction of data with the lowest classification confidence and sends them to the providers for classification. The providers use their private classification models to label the data and send the resulting classifications back to the requestor. The requestor aggregates the received classifications with the results of its own classifier and determines the final classification of the data. The aggregation mechanism used in our implementation is majority voting with a preference for own classification in case of ties. A certain disadvantage of the data classification strategy for the requestor is that by specifying the data to be classified the requestor discloses some information about its own data.

In order to measure the private knowledge loss using the KL measure, an additional step is needed when using the data classified strategy. The labelled data provided by the providers in response to the requestor's queries are first used to train a Bayes classifier; this classifier is then used to calculate the amount of private knowledge disclosed.

7 Experiments

In order to validate our approach and to obtain quantitative data on the behaviour of the proposed metrics and strategies, we have designed a set of experiments on a synthetic domain.

7.1 Experiment Domain and Setting

The feature space is composed of two features; each feature can assume a value from $\{0, 1, \ldots, 9\}$. The feature vectors belong to one of ten possible classes with equal probability. Data samples for each class are drawn from a Gaussian distribution (restricted to the feature space) defined by its mean and a fixed unit variance. The distribution means are selected randomly with a uniform probability distribution from the whole feature space. As a result, identical feature vectors with different classifications can appear in the domain. The situation is illustrated in Figure 2.

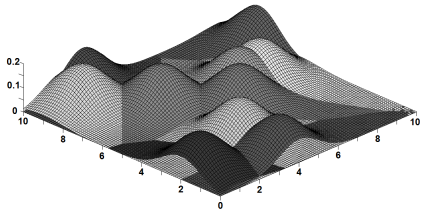

Fig. 2. The domain used in the experiments. Each Gaussian generates data samples from a different class.

The community of agents in the experiments comprises ten identical agents $A_0, A_1, \ldots A_9$. Each agent employs one of the two cooperation strategies (see Section 6) to classify its own task data with the help of the classification knowledge obtained from all other agents. Each agent has its private classification model trained on hundred data samples (D_i) and it has another thousand data samples to classify as its task data (T_i). All presented results are averages over one hundred runs of the experiment with identical settings.

KL divergence as well as the naïve Bayes classification works well only with positive distributions, i.e. each combination of features and classification is considered possible (there are no zeros in the distribution). In order to achieve this property in our experiments, we initialize the frequencies representing the Naïve Bayes classifier as explained in Section 5.2 with ones. However, these extra ones are kept only once in the process of model merging.

7.2 Results for the Model Sharing Strategy

The main parameter of the model sharing strategy is the portion of the training data the provider uses to train a degraded classifier that it sends to the requestor. The dependency of individual measures on this parameter are depicted in the upper row of graphs in Figure 3.

Classification Accuracy Classification accuracy measure plotted in Figure 3(a) shows how model parts from other agents merged together with the original model of the requestor agent A_0 improve requestor's classification accuracy. The increase in prediction accuracy was expected – the bigger the part of provider agents' models merged into the model of the requestor, the more information about the domain the resulting model captures. If the requestor A_0 receives models learned only from 5% of providers' data, the accuracy of its classification is approximately 61%. The accuracy increases quickly at the beginning but the improvement tails off closer to sharing full models; this is because the private classification model is already well approximated at this point.

Private Knowledge Loss In this strategy, the provider agents disclose their private knowledge in the form of partial classification models – see the results for

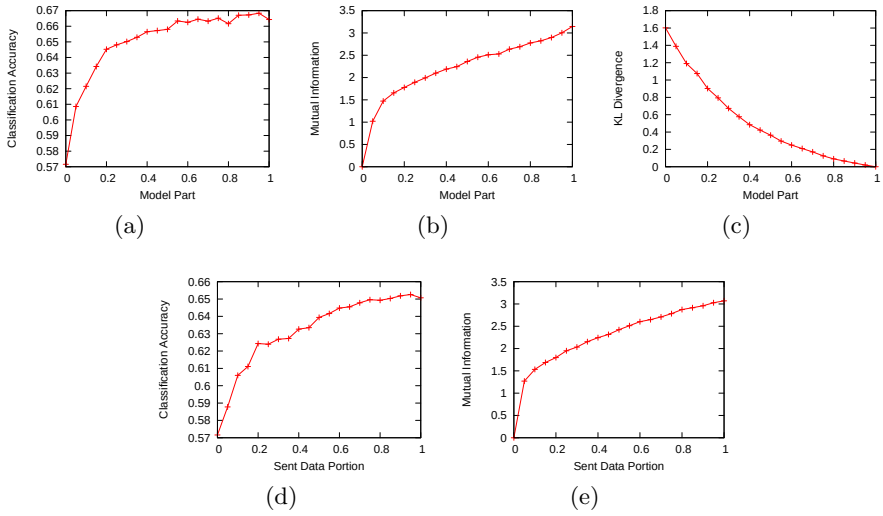

Fig. 3. Classification precision of requestor A_0 using the model sharing strategy and the private knowledge loss of agent A_1 responding to requestor's queries according to various measures for the model sharing (a,b,c) and the data classification (d,e) strategies.

the individual measures below. Note that the requestor agent does not disclose any information at all.

Mutual Information The mutual information is measured between the original private model of the provider and the (degraded) model sent to the requestor. The data used to sample the distribution $P(X, Y)$ from formula 3 are the complete training set the provider used to obtain its private classifier. The dependency of the mutual information between the models and the portion of the data used to train the shared model is depicted in Figure 3(b). Even though it is measured on a different agent, the trend is similar to the classification improvement of the requestor. It means that the mutual information metric accurately describes the amount of useful information contained in the sent model; more data used for creating the degraded model implies higher private knowledge disclosure.

KL-Divergence Figure 3(c) depicts the trend of KL divergence, which measures the difference between the disclosed model and the private one. The trend is in agreement with expectations – initially, increasing the amount of data on which the degraded model is created contributes strongly to narrowing the difference between the provider's private model and the sent model; the contribution decreases when the degraded model approaches the provider's private model (i.e. when the portion of the training data used approaches one).

7.3 Results for the Data Classification Strategy

The main parameter of the data classification strategy is the percentage of the task data with the smallest classification confidence the requestor sends to the providers for classification. The experimental results concerning this strategy are summarized in the bottom row of graphs in Figure 3.

Classification Accuracy Figure 3(d) confirms the basic hypothesis that the classification accuracy can be improved using the data classification strategy – the accuracy grows clearly with the increased amount of shared testing data of the requestor. The irregularities in the trend are caused by fluctuations of the classification precision between individual runs – a smoother graph would be obtained for a higher number of runs or a large data set. The improvement through collaboration is largest when the data on which the requestor has a very low confidence are sent to the providers for classification. The increase in accuracy diminishes when the agent requests classifications for the majority of its testing data; this is because then the requestor already has a good chance of predicting the class correctly.

Private Knowledge Loss In contrast to the model sharing strategy, both the requestor and the providers lose private knowledge in this case. The requestor A_0 loses private knowledge about its task data while the provider agents lose information about their private models.

The amount of information lost by the requestor can be quantified by the number of examples it sends to the other agents for classification. The loss for the providers can be expressed by the proposed measures.

Mutual Information The plot in Figure 3(e) depicts the loss of knowledge about providers' private models caused by responding to requester's queries. The loss of the private knowledge of providers increases with the number of queries answered, fast at the beginning and slower towards the end when the requestor already has enough information to reconstruct the providers' models accurately.

KL-Divergence We do not present a plot of KL divergence metric for this strategy because the metric proved unsuitable for the data classification strategy. KL divergence computes the distance between two probability distributions realized by the corresponding classifiers. In order to use KL divergence, we first need to reconstruct the classifier from the classified samples sent. The problem arises from the fact that different distributions can correspond to identical classifications. If a feature vector can belong to multiple classes in the ground truth, the original classifier can learn this fact but it always outputs only the most probable class. A classifier trained only from sampling the original classifier will have the probability of classifying to the dominating class equal to one, and the probabilities of classifying to the other classes equal to zero. As a result, the KL divergence can grow even though the similarity of produced classifications increases.

7.4 Strategy Comparison

Both the evaluated strategies are usable for improving classification accuracy of the requestor agents. Starting from the same base level (57%), the model sharing strategy achieved up to 67% accuracy while the data sharing reaches only 65% even if all the thousand task data samples are shared. On the other hand, the model sharing strategy has to reveal the whole model of the provider agent (instead of classified samples) in order to achieve the peak accuracy. We compare loss of the private knowledge about the task data and the private knowledge contained in classifiers.

The information about the task data is communicated only in the case of data sharing strategy. The requestor agent sends a portion of its testing data given by the strategy parameter to all provider agents. The strategy parameter represents how much of the private testing data will be disclosed.

The knowledge about private classification model is disclosed in both strategies. The measure suitable for measuring the amount of private knowledge revealed in both of them is the mutual information metric. The graphs for both strategies are almost identical. According to that, sending model trained on k samples form an agent training set reveals as much private knowledge about the complete model as answering $10 * k$ queries for classification of other agents task data. The size of the task data sets is ten times bigger than the size of the training sets.

The inapplicability of KL divergence as a sound private knowledge loss measure for the data classification strategy indicates that the knowledge exchanged in the strategy does not allow to fully reconstruct a provider's model, in particular its classification confidence. This is not the case for the model sharing strategy where the confidence information is disclosed as part of the exchanged partial models.

8 Conclusion

We have analyzed the trade-offs between improving classification accuracy and losing private classification knowledge in the multi-agent setting. Reasoning about such trade-offs is of high relevance whenever multiple competitive parties want to cooperate in order to improve their own performance. We believe such situations are common in real-world environments, including various business sectors (e.g. banking, insurance etc) or areas of international cooperation.

Understanding that cooperation cannot arise unless individual parties can reason about losses and benefits entailed by such cooperation, we have designed a set of measures and a benchmark task on which the measures can be validated. Afterwards, we have presented two elementary strategies consisting of a set of operations which are likely to be the building blocks of any multi-agent strategy aimed at improving classification accuracy through agent cooperation. The first strategy is based on sharing partial classification models; the other strategy employs consulting other agents' opinions about the classification of specific

examples. We have evaluated the strategies on a synthetic domain and compared their respective advantages and disadvantages. The results obtained are in agreement with our theoretical expectations and indicate that the framework developed could be used for building semi-cooperative data mining systems.

The work presented is, in any case, only a first step towards enabling more complex strategies e.g. involving reciprocity or payment for the private knowledge disclosed. In a longer term, we aim to design fully autonomous agents that can automatically reason about the benefits and costs of individual knowledge sharing operations in the context of semi-cooperative predictive data mining.

9 Acknowledgments

We gratefully acknowledge the support of the presented research by Office for Naval Research project N00014-09-1-0537 and the Research Programme No. MSM6840770038: Decision Making and Control for Manufacturing III by the Ministry of Education of the Czech Republic.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proc. of the ACM SIGMOD Conference on Management of Data, ACM Press (May 2000) 439–450
2. Kantarcioğlu, M., Jin, J., Clifton, C.: When do data mining results violate privacy? In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2004) 599–604
3. Clifton, C., Kantarcioglu, M., Vaidya, J.: Defining Privacy for Data Mining. In: National Science Foundation Workshop on Next Generation Data Mining. (2002) 126–133
4. Bonchi, F., Saygin, Y., Verykios, V., Atzori, M., Gkoulalas-Divanis, A., Kaya, S., Savas, E.: Privacy in Spatiotemporal Data Mining. Mobility, Data Mining and Privacy: Geographic Knowledge Discovery (2008)
5. Natwichai, J., Li, X., Orlowska, M.: Hiding Classification Rules for Data Sharing with Privacy Preservation. LECTURE NOTES IN COMPUTER SCIENCE **3589** (2005) 468
6. Verykios, V.S., Gkoulalas-Divanis, A.: A Survey of Association Rule Hiding Methods for Privacy. In: Privacy-Preserving Data Mining. Springer US (2008)
7. Bertino, E., Lin, D., Jiang, W.: A Survey of Quantification of Privacy Preserving Data Mining Algorithms. In: Privacy-Preserving Data Mining. Springer US (2008)
8. Bertino, E., Fovino, I., Provenza, L.: A Framework for Evaluating Privacy Preserving Data Mining Algorithms*. Data Mining and Knowledge Discovery **11**(2) (2005) 121–154
9. Franzin, M., Rossi, F., Freuder, E., Wallace, R.: Multi-Agent Constraint Systems with Preferences: Efficiency, Solution Quality, and Privacy Loss. Computational Intelligence **20**(2) (2004) 264–286
10. van der Krogt, R.: Privacy loss in classical multiagent planning. Intelligent Agent Technology, IEEE / WIC / ACM International Conference on **0** (2007) 168–174
11. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)