Collective vs Independent Classification in Statistical Relational Learning

AMIT DHURANDHAR
University of Florida
and
ALIN DOBRA
University of Florida

Statistical Relational Learning (SRL) addresses the problem of performing probabilistic inference on data instances that are correlated. Collective classification is an important SRL task, in which related data instances are classified simultaneously as opposed to independently which is done in independent Machine Learning. In several studies conducted in the last decade, it has been shown that collective classification, by exploiting relational information, tends to outperform independent (supervised) classification on various relational datasets. However, owing to their ability to exploit relational information collective classification algorithms are invariably more complex than their independent counterparts. Moreover, independent classification algorithms have been around for a considerably longer period of time and hence have been thoroughly studied both theoretically and experimentally. Consequently, a natural question that arises is: Under what circumstances should we perform collective classification? Previously, it had been argued that on relational datasets which exhibit high auto-correlation between related instances linked through arbitrary paths in a relational data graph, collective classification is superior. In this paper however, we partition the feature space into (a) Direct features – links from an object type (which contains the class attribute) to itself (b) Indirect features – the remaining set of features (i.e. other links and object types) and argue that high auto-correlation between instances linked through Direct features and low/medium auto-correlation between instances linked through Indirect features is essential if collective classification is to significantly outperform independent classification. Moreover, based on this argument and depending on the setting (i.e. level of auto-correlation) we motivate simple baseline classification algorithms which can be used as yardsticks to evaluate the more sophisticated collective classification algorithms. We validate our arguments by performing experiments on synthetic and real datasets.

Categories and Subject Descriptors: H.2.8 [**Data Management**]: Data Mining

General Terms:
Additional Key Words and Phrases: Statistical Relational Learning, collective classification, auto-correlation

A. Dhurandhar, University of Florida, Gainesville, Fl-32611, USA.
A. Dobra, University of Florida, Gainesville, Fl-32611, USA.

1. INTRODUCTION

The study and development of classification algorithms covers a major portion of the research that is conducted in Machine Learning and Data Mining. Traditionally, the data is considered to be drawn independently and identically from some distribution (i.i.d.). Almost all the theory developed regarding the study of learning algorithms [Vapnik 1998] is based on this fundamental assumption. Though the theory is elegant, the fundamental assumption on which it is based is rarely satisfied in reality. In real life more often than not there exist correlations between data instances which violates the i.i.d. assumption. Moreover, a sizable amount of data is stored in relational databases where instances are inter-dependent by design. The need to model such dependencies between instances has led to the emergence of a sub-area in Machine Learning, Statistical Relational Learning (SRL). Research in SRL is directed towards modeling uncertainty in relational data. The primary focus of this sub-area is to forgo the i.i.d. assumption that has been made almost throughout Machine Learning research.

Collective classification is one of the important tasks that has been closely looked at in SRL research [Chakrabarti et al. 1998; Jensen et al. 2004; Neville and Jensen 2003; Taskar et al. 2001]. In collective classification, related instances (or objects to be classified)[1] are classified not just based on their own set of attribute values but also based on the attribute values and class labels of the related instances. An example application is trying to classify hyperlinked webpages based on their topic. In this situation, not just the words in a particular webpage but the topics of webpages that are linked to it might significantly aid in determining its own topic. This was shown in [Chakrabarti et al. 1998], where the authors concluded that collective classification was extremely useful in categorizing webpages using hyperlink information. Independent classification algorithms could not exploit this information and hence, underperformed by a significant margin. In [Getoor et al. 2004] the authors portrayed the use of collective classification in tracking contagious diseases. Here again collective classification was superior to independent classification. There have been several other studies [Neville and Jensen 2003; Taskar et al. 2001; Richardson and Domingos 2006; Getoor and Taskar 2007] which have depicted the efficacy of collective classification over independent classification. However, since collective classification algorithms are equipped to model dependencies in relational data they are invariably more complex than their independent counterparts. For example, in Relational Markov Networks [Taskar et al. 2002] which is a well known collective classification model, parameter estimation (i.e. learning) becomes exponentially (in the size of the cliques connecting labeled instances) more expensive than in the i.i.d. setting [Getoor and Taskar 2007], since the joint probability no longer factorizes. Consequently, approximations have to be used. These approximations are also needed in other prevalent relational models [Richardson and Domingos 2006; Neville and Jensen 2007] for them to be practical. Another example is a Probabilistic Relational Model [Getoor et al. 2004] where extra attributes have to be added to model the relationship between two or more instances. This added complexity in collective models arises since the state space in modeling the

[1]We will interchangeably use the terms, instances and objects.

joint distribution over relational data is much larger compared to i.i.d. data. In particular, if we have d attributes with each attribute taking v values, a dataset of size N where the size of the largest connected component is $m \in \{1, ..., N\}$, then the state space in the i.i.d. setting $O(v^d)$ is much smaller than the state space in the relational setting $O(v^{md})$ since, m is usually $>> 1$. Another advantage of independent classification is that independent classification algorithms have been around for a considerably longer period of time and hence have been thoroughly studied both theoretically and experimentally leading to better understanding.

Given these trade-offs between collective classification algorithms and independent classification algorithms, it is important to identify situations in which one classification paradigm would be more desirable than the other. A study that is notable in this regard is [Jensen et al. 2004] in which the authors inferred through empirical studies that collective classification outperforms independent classification when the auto-correlation (i.e. correlation between values of the class attribute) between linked instances in the data graph is high. They found that as the auto-correlation increases and especially when the proportion of known labels in the test set is about 30% or more, collective classification has significantly superior performance. The collective models that they used to arrive at the above conclusion consisted of a single object type with individual objects of this type linked to each other directly (i.e. not through other objects and corresponding links. e.g. one paper citing another paper). As is mentioned in [Jensen et al. 2004] auto-correlated objects are many times connected/linked through objects and links of other types. For example, in a citation database different research papers can be linked to each other through common authors or in a movie database different movies may have the same director. However, this scenario is not captured by the collective models they consider. Based on the behavior of these restrictive collective models they claimed that high auto-correlation between objects linked with each other either directly or through paths consisting of objects (and links) of other types, is the primary motivation for using collective classification as opposed to independent classification. In this paper we argue that this claim is an overgeneralization based on their observations and provide alternative necessary conditions for collective classification to outperform independent classification given the above trade-offs. These conditions naturally motivate baseline algorithms that can be used as yardsticks to evaluate the state-of-the-art collective classification algorithms. We validate all our arguments by performing experiments on synthetic and real datasets. Details with regards to these contributions are as follows:

(1) We provide necessary conditions for collective classification to significantly outperform independent classification. These conditions are motivated based on our argument of the ability of these two classification paradigms to be able to accurately model the available information in relational datasets.
(2) We motivate simple baseline classification algorithms where the choice of a particular baseline algorithm is governed by the specific situations characterized in (1). The suggested baseline algorithms are, a) an independent classification algorithm for some of these situations and b) a Nearest Neighbor algorithm adapted to the relational setting for the other situations.
(3) We conduct empirical studies on synthetic and real datasets to validate our

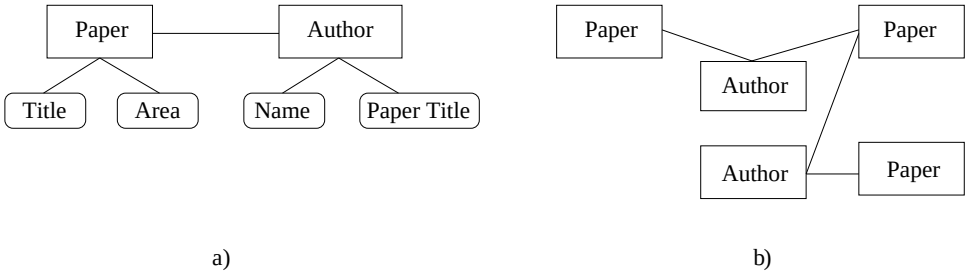

Fig. 1. (a) represents a relational schema with object types, *Paper* and *Author*. The relationship between them is many-to-many. The rounded boxes linked to these object types denote their respective attributes. (b) is the corresponding data graph which shows authors linked to the papers that they authored or co-authored.

arguments. Our observations on the real datasets are qualitatively similar to previously published results and we explain how these observations are consistent with our argument.

Note that both the necessary conditions and the suggested baselines are applicable after one has performed feature selection. This paper is not concerned with how one chooses the appropriate set of features (e.g. the right aggregation function, the right attributes etc.) but rather with how one chooses the right classification paradigm and the appropriate baseline following this process.

The paper is organized as follows: In Section 2 we define basic terms and concepts. In Section 3 we provide necessary conditions for collective classification to outperform independent classification. Before stating these conditions we explain the logic in arriving at them. Moreover, we also argue that the conclusions in [Jensen et al. 2004] were an overgeneralization based on their observations. The necessary conditions stated in Section 3 lead to simple baseline classification algorithms which we state in Section 4. In Section 5 we validate our arguments by performing empirical studies on synthetic and real datasets. In Section 6, we discuss the effects of data graph transformations and increasing Indirect features has on the claims made in this paper. In Section 7, we discuss limitations of our work which lead to interesting avenues that need further investigation in the future. We summarize the major developments in the paper in Section 8.

2. PRELIMINARIES

In this section we define concepts and terminology that is used throughout the paper.

Relational data: Relational data consists of objects and the relationships between these objects are called links. Each object and link have a *type* associated with them. Objects or links of the same *type* have the same set of attributes. Relational data is represented at the *type* level by a graph which is called a relational schema. Relational data represented at the individual object and link level as a graph is called a relational data graph [Neville 2006], wherein the vertices are the objects and the edges are the links. An example relational schema and the corresponding

Paper

Title	Area
paper1	AI
paper2	AI
paper3	Graphics

Author

Name	Paper Title
author1	paper1
author1	paper2
author2	paper2
author2	paper3

Fig. 2. Relational database representation of the relational dataset in Figure 1b. The table on the left contains objects of type *Paper* and the table on the right contains objects of type *Author*. The attribute Title is a primary key in *Paper* and the attribute Paper Title is the corresponding foreign key in *Author*.

data graph (i.e. the actual dataset) are shown in Figures 1a and 1b respectively. The relational schema has 2 object types namely; *Paper* and *Author*. The data graph shows 2 author objects linked to paper objects that they authored or co-authored.

The most prevalent representation of relational data is in Relational Database Management Systems (RDBMS) where object and link information is stored in tables. A single table stores objects or link information of a particular *type*. The columns of such a table denote the attributes associated with the particular *type* while each row stores information of each individual object (or link) of that *type*. This is shown in Figure 2, which represents a relational dataset that identical to the one represented by the data graph 1b. Though the data is stored in separate tables to avoid redundancy, the information in these tables can be put into a single large table called the Universal table [Malvestuto 1989]. This procedure of combining smaller tables to form larger tables is called a Database Join (or just a Join)[2]. Usually a Join is between the primary key in one table and the foreign key in another table. A primary key is an attribute/set of attributes that uniquely identify a row in their table (e.g. Title in *Paper*) and a foreign key is an attribute/set of attributes that uniquely identify a row in a different table (e.g. Paper Title in *Author* uniquely identifies rows in *Paper*).

In our descriptions throughout the paper we will interchangeably refer to either of the above two representations as deemed appropriate.

Data Heterogeneity: Probabilistic models over relational data [Getoor and Taskar 2007] are used to handle uncertainity in relational domains. One of the main challenges in learning over these domains is data heterogeneity, something we do not have to deal with in i.i.d. domains. A single object of a certain type may be linked to multiple objects of some other type. An example of this can be seen in Figure 1b. One of the papers in this data graph has two authors. Consequently, this paper is associated with 2 values of the attribute Name. A popular solution to this

[2]Joins are of different types but their discussion is unnecessary for this paper.

problem is to aggregate the values of the attribute into a single value. For instance in our example we can use an aggregation function such as Exists() which takes as parameters the names of the authors and returns True if that paper has a particular author and False otherwise. If numeric attributes are present such as Age or Salary, aggregation functions such as Average(), Mode(), etc. can be used. Since, an aggregation function applied to an existing feature gives rise to a new feature, the choice of aggregation function is an important aspect of feature selection.

Independent Classification (IC): This refers to independent classification where instances are classified based only on the values of their attributes. For example in Figure 2, the objects Paper have intrinsic attributes (i.e. attributes belonging to the same object type such as *Paper*) Area, Title and the relational attribute (i.e. attributes belonging to a related object type such as *Author*) Name (Paper Title is the same as Title and hence we do not include it.). If a Paper object is to be classified into the appropriate research area (i.e. Area is the class attribute), it is classified based only on its own values of Title and Name. Relational Bayes Classifier, Relational Probability Trees, SVM are examples of IC algorithms.

Collective Classification (CC): In collective classification the class labels of multiple instances are inferred simultaneously, assuming dependencies between these instances. Thus, the class label of a particular instance depends on the class labels and sometimes even attributes of the other related instances and not just on its own set of attributes. Consider the same example as the one for independent classification where Paper objects have three attributes Area (the class attribute), Title and Name. For independent classification we predicted the research area of a paper based only on its title and the name of its author. In collective classification, besides the papers own attributes we will use information regarding the research areas of other papers that have the same author. Thus, the Area attribute of one paper is predicted by its own attributes Title and Name and in addition the Area attributes of other papers with the same value of Name. Using this additional piece of information can sometimes significantly enhance classification accuracy [Chakrabarti et al. 1998; Getoor and Taskar 2007]. For an in dept survey of CC alongwith a discussion of popular inference procedures please refer to [Sen et al. 2008].

Relational auto-correlation: Relational auto-correlation measures the strength of statistical dependencies between values of a single attribute on related/linked instances. In this paper as in previously published articles [Angin and Neville 2008; Neville 2006] we measure relational auto-correlation using the Pearson's Contingency Coefficient given by,

$$\rho = \sqrt{\frac{\chi^2}{N + \chi^2}}$$

where χ^2 is the Chi-square statistic and N is the sample size. In this paper whenever we mention auto-correlation we imply relational auto-correlation.

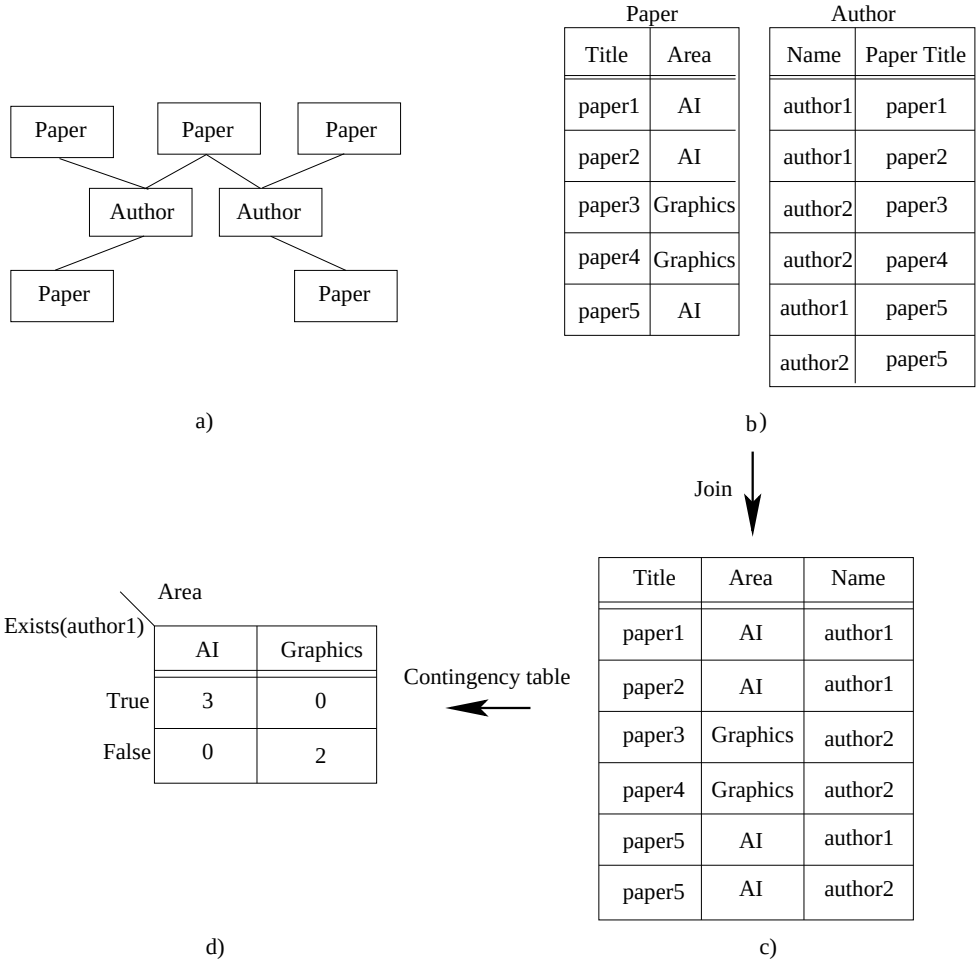

Fig. 3. (a) shows an example data graph corresponding to the relational schema in Figure 1a. (b) is the relational database representation of the data graph in (a). (c) is the table formed by joining the tables in (b). (d) is a contingency table formed from (c) given that we want to classify papers by their respective areas. This table can be used to train an independent classification algorithm.

3. COLLECTIVE VS INDEPENDENT CLASSIFICATION

In this section we provide necessary conditions for collective classification to outperform independent classification. Before stating these conditions we explain the logic in arriving at them. Moreover, we argue that the conclusions in [Jensen et al. 2004] were an overgeneralization based on their observations. We explain our position below where we argue that high auto-correlation in a relational dataset does not necessarily imply that collective classification will be superior to independent classification.

3.1 Scenario 1: High auto-correlation but CC and IC comparable

Consider the example in Figure 3 where we have a dataset consisting of five papers and two authors. In this example we want to classify papers into their appropriate area, AI or Graphics. The data graph and the corresponding database representation of this dataset is shown in Figures 3a and 3b respectively. We observe from these figures that one of the papers has two authors. Usually in such cases an aggregation function is used to obtain a single value. In this case we use the boolean function Exists(Name) which takes as input author name and outputs True if that particular author is one of the authors for paper in question. If not, the function outputs False. For example, the value of Exists(author1) is True for paper1 but False for paper4. Again, note that the choice of aggregation function is part of the feature selection process (since Exists(Name) is feature) which is a preprocessing step to the contributions made in this paper. It is easy to see that the auto-correlation of linked instances through this feature is high for this dataset. Given this any reasonable collective classification algorithm should perform well on this kind of dataset. The important question now is: how will an independent classification algorithm perform? To answer this question we form a contingency table from the given dataset. The columns of a contingency table denote values of the class attribute (output). In our example AI and Graphics are values of Area. The rows represent the values of other attributes which may include all attributes except the class attribute or a proper subset of the other attributes that may be chosen after feature selection (Exists(Name) in this case). The individual cells (intersection of rows and columns) in a contingency table contain counts of the number of inputs that have an output value corresponding to the row and column that form the respective cell. In the contingency table in Figure 3d the cell corresponding to input True and output AI has a count of 3 since author1 has authored/co-authored three papers all of which are in AI. By the same reasoning the cell corresponding to input False and output Graphics has a count of 2. From this contingency table we see that the attribute Exists(Name) is highly *cross-correlated* with the class label Area. Consequently, any reasonable independent classification algorithm should be able to exploit this information and perform well on this kind of dataset. Hence, both classification paradigms would have favorable performance, though the data exhibits high auto-correlation.

In general if we have objects to be classified (e.g. Papers) connected through arbitrarily long paths in a data graph consisting of objects of other types, with the data exhibiting high auto-correlation through these paths, then both classification paradigms can perform well. In other words, the auto-correlation through the above mentioned paths being high, implies the existence of attributes that are highly cross-correlated with the class label. With these attributes we can form a contingency table and train any reasonable independent classification algorithm which should perform comparable with any reasonable collective classification algorithm. Thus, the available information in such datasets can be made available to classification algorithms in both paradigms in a manner that they can exploit.

The collective classification models and datasets that were considered in [Jensen et al. 2004] consisted of a single object type with links connecting objects of the same type. From our discussion above we can see that the results that they ob-

served in this restricted setting cannot be generalized to the above scenario i.e. high auto-correlation in a relational dataset does not necessarily imply that collective classification will be superior to independent classification.

3.2 Scenario 2: High auto-correlation and CC superior to IC

Consider the example in Figures 4 and 5, where we have a relational dataset consisting of four papers, two authors and citation information. We can see from the Figure 5 that the data has high auto-correlation through the link type *Cites*, since papers linked through this type are in the same area. On the other hand, the auto-correlation through the type *Author* is low, since papers having a common author are in different areas. Since, the link type *Cites* relates two objects of the same type, with this relationship carrying decisive information about the class labels (and not individual or set of attributes), it is not clear how to include this information in a contingency table. One possible solution can be to introduce a new attribute such that all Paper objects linked through Cites have the same value within each group and different values across groups. For example, we introduce a new attribute say Z. In Figure 5, z_1 is its value for paper1 and paper2 and z_2 is its value for paper3 and paper4. This process however, of introducing new attributes (or objects) and analyzing their effects is a research problem in itself and if done in an automated fashion, is part of the more general problem known as Statistical Predicate Invention [Kok and Domingos 2007]. In short, it is not clear how this relationship information between papers can be incorporated into a contingency table without causing unnecessary side effects and be made available to an independent classification algorithm. Moreover, most researchers would consider a technique which employs such a modeling as a CC technique rather than an IC technique, by its use of relational information between directly linked instances. Some of the most prominant variants of such CC techniques/algorithms use a base classifier and an iterative procedure to do prediction. Examples of such techniques, are Iterative Classification Algorithm (ICA) and Gibbs sampling [Sen et al. 2008]. We can as in scenario 1 form a contingency table with just the attributes Name and Area but in this case the Name attribute is not nearly as discriminative and hence the independent classification algorithm trained on this data will perform poorly.

In general, when objects to be classified are linked to each other directly (e.g. paper cites paper) and not through paths consisting of objects of other types (e.g. papers linked through a common author) and the auto-correlation through these direct links is high, then collective classification can improve classification accuracy. On the other hand, there is no straightforward way in which independent classification might able to exploit this information and enhance classification accuracy.

The collective classification models and datasets that were considered in [Jensen et al. 2004] were qualitatively similar to this scenario where the auto-correlation was present through links connecting objects to be classified (which are of the same type). In this scenario our argument is consistent with the observations in [Jensen et al. 2004], where the authors concluded that high auto-correlation corresponds to superior performance of collective classifiers.

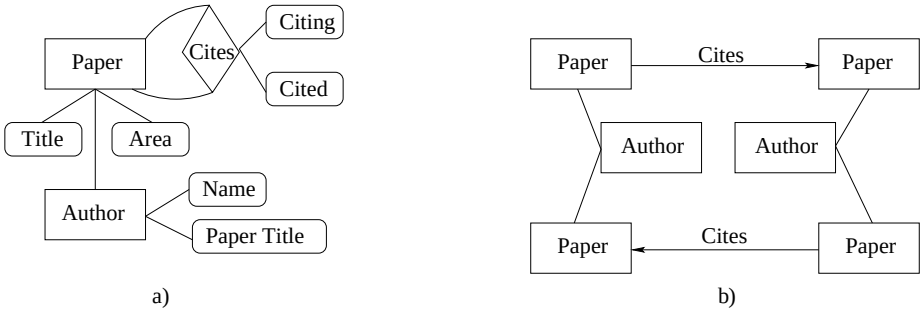

Fig. 4. (a) represents a relational schema with object types *Paper* and *Author* and link type *Cites*. All relationships are many-to-many. The rounded boxes linked to these types denote their respective attributes. (b) is the corresponding data graph which shows authors linked to the papers that they authored or co-authored and papers cited by other papers (arrows point to the cited paper from the paper that cites it).

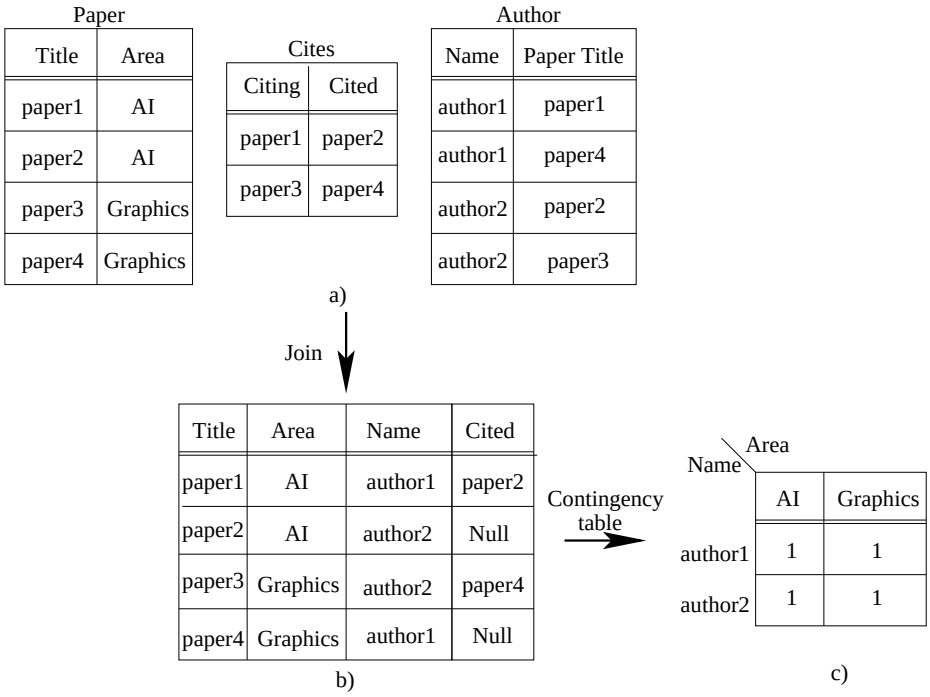

Fig. 5. (a) is the relational database representation of the data graph in Figure 4b. (b) is a table formed by joining the three tables in (a). (c) is a contingency table formed from (b) given that we want to classify papers by their respective areas. This table can be used to train an independent classification algorithm.

3.3 Predicted comparative performance of CC and IC

Based on the above discussions we will now provide necessary conditions for CC to

AD↓, AID→	High	Medium	Low
High	S	B	B
Medium	S/W	S	B/S
Low	S/W	S	S

Table I. AD and AID imply level of auto-correlation through Direct features and Indirect features respectively. B stands for better, S stands for same and W stands for worse. B/S stands for slightly better or same and S/W stands for same or slightly worse. The entries in the table indicate the performance of CC when compared with IC for different levels of auto-correlation through Direct and Indirect features based on our discussion of the two scenarios in Section 3.

significantly outperform IC. However, to make the exposition of these conditions clear, we first formally define two types of features which partition the feature space.

Consider a relational schema where object type T has attributes $t_1, ..., t_n$ and without loss of generality let t_1 be the class label. Let T be linked to m other object types $S_1, ..., S_m$[3] where S_i $i \in \{1, ..., m\}$ has n_i attributes $s_{i,1}, ..., s_{i,n_i}$. Let $R_1, ..., R_k$ be binary relations that link objects with object type T to other objects with the same object type and where R_i $i \in \{1, ..., k\}$ specifies these relationships through the attributes $r_{i,1}, r_{i,2}$. Given this setup, we define two types of features namely; Direct features and Indirect features.

(1) *Direct features:* Any function (e.g. subset()) on the set $\{(r_{1,1}, r_{1,2}), ..., (r_{k,1}, r_{k,2})\}$ is a Direct feature. An example of Direct feature is (Citing,Cited) attributes of the relation *Cites* in Figure 4a.

(2) *Indirect features:* Any function (e.g. subset(), aggregation functions) on the set $\{t_2, ..., t_n, s_{1,1}, ..., s_{1,n_1}, ..., s_{m,1}, ..., s_{m,n_m}\}$ is an Indirect feature. An example of Indirect features is the attribute Name of the object type *Author* in Figure 4.

Note that the Indirect features defined above are significantly different from *identifiers* defined in [Perlich and Provost 2006]. As defined in [Perlich and Provost 2006] identifiers are essentially categorical attributes having at least a specified number of distinct values on which two tables can be joined. An example of an identifier is a primary key in a table. This is different from Indirect features since Indirect features have no constraint on the number of distinct values an attribute must have and neither is it required that the attribute (or its equivalent) be present in two different tables.

Given the definitions of the two kind of features and based on the explanation of the two scenarios described before, the *necessary conditions* for collective classification to outperform independent classification are:

(1) auto-correlation through Direct features is high and

(2) auto-correlation through Indirect features is medium/low.

Since the level of auto-correlation in a relational dataset through each of these two types of features (after feature selection) can be either high, medium or low

[3] T could be directly linked to each of these object types or could be linked through other object types. For example, T could be linked to S_2 through S_6 which is directly linked to T.

AD↓, AID→	High	Medium	Low
High	IC	DRN	DRN
Medium	IC	IC	IC
Low	IC	IC	IC

Table II. Suitable baselines for varying levels of auto-correlation are suggested above. AD and AID imply level of auto-correlation through Direct features and Indirect features respectively. DRN denotes the Direct Relational Neighbor model and IC denotes any reasonable (which depends on the application. e.g. for text categorization SVM or Naive Bayes may be used) independent classification model. The entries in the table indicate suggested baseline models that can be used as yardsticks to evaluate state-of-the-art CC models.

(absence of any of these 2 features is qualitatively equivalent to the case when auto-correlation is low through that feature)[4] we have 9 possible categories in which a dataset can lie. These 9 categories and the corresponding relative performance of collective classification w.r.t. independent classification as can be inferred from our preceding discussions, are given in Table I. In the cases where auto-correlation through Indirect features is high and through Direct features is medium/low we expect independent classification to perform slightly better than collective classification since the choice of independent classification algorithms with arguably superior implementations is greater than their collective counterparts (larger the hypothesis space lower the error [Vapnik 1998]). Moreover, learning and inference in most of the state-of-the-art collective classification algorithms is performed using approximate techniques (e.g. Gibb's sampling and other Markov chain Monte Carlo techniques for inference, pseudo-likelihood for learning [Getoor and Taskar 2007]) which can lead to higher error. The prediction for the other cases directly follows from our arguments stated before.

4. BASELINE CLASSIFIERS

In the previous section we identified 9 cases depending on the level of auto-correlation through Direct and Indirect features in a relational dataset. The comparative performance (based on our discussion) of CC and IC for each of these cases is shown in Table I. Looking at this table we can infer that CC outperforms IC in 2 of the 9 cases that is when auto-correlation through Direct features is high and auto-correlation through Indirect features is medium/low. In the remaining 7 cases IC is comparable in performance to CC. In these cases independent classification algorithms can prove to be a suitable baseline for the more sophisticated collective classifiers. For the 2 cases when CC is predicted to be significantly superior to IC we suggest using a specific version (described below) of the Relational Neighbor (RN) model [Macskassy and Provost 2003; 2007]. The reason we suggest this variant is that, we believe it is an exceedingly simple yet effective model for these specified cases; something that a good baseline should possess.

The NN model we suggest adapted to the relational setting will classify objects based on the class labels of other objects of the same type that are linked through

[4]If any of these two types of features is not present in the particular dataset then the case corresponds to the auto-correlation being low through that feature since this low auto-correlation implies that the feature is useless for classification and hence its presence or absence in the dataset would not greatly affect the classification accuracy (since feature selection will remove this feature).

Direct features. The model will assign a class label to an object which is most numerous amongst its neighbors. Neighbors may include objects one link away (i.e. immediate neighbors) or up to multiple links away (e.g. up to 2 links away) or exactly some number of links away (e.g. exactly 2 links away). This choice depends on the characteristics of the particular relational dataset. If the class labels of all neighbors are not known it will classify based on the class labels of neighbors that are known. Thus, the method requires no learning and is simple to apply. However it, requires that some class labels be accurately known at the start and the auto-correlation through Direct features be high, for it to perform well.

As mentioned before, a more general (and a more complicated) form of such a model was previously suggested as baseline [Macskassy and Provost 2003; 2007]. We say more general, since RN could classify an instance based on the class labels of its neighbors linked through both Direct and Indirect features while our NN model classifies only through Direct features. We will refer to our model as Direct Relational Neighbor classifier (DRN). They suggested RN to be a baseline irrespective of the characteristics (i.e. auto-correlation through the various features) of the particular relational dataset in question. In this paper however, we suggest a simpler version of this model namely; DRN as a baseline only for the cases when auto-correlation through Direct features is high and auto-correlation through Indirect features is medium/low, since only if the auto-correlation through Direct features is high, will DRN perform well. For the other cases we suggest IC models to be suitable baselines. The reasoning behind these suggestions is as follows: when auto-correlation through Indirect features is high state-of-the-art IC models will perform well irrespective of the level of auto-correlation through Direct features. When the auto-correlation through both Direct and Indirect features is low/medium, which implies that the available information is not particularly useful for classification, RN and IC models should perform equally poorly. The need to have accurate information about the class labels of a fraction of the instances is a limitation of the NN approaches if they are to perform well. IC which does not suffer from this limitation, is thus a better choice as baseline in the suggested cases. Moreover, we have more choice when it comes to choosing IC models (since there are many) and we can choose the appropriate model as baseline depending on the application. For example Bayesian Networks and Support Vector Machines are IC models which have been shown to be successful in spam filtering and text categorization [Joachims 2002] respectively and hence can be reasonable baselines in those domains when the auto-correlation through Direct features is low/medium.

Based on the above argument the baseline models we suggest for the 9 cases are given in Table II.

5. EXPERIMENTS

In this section we empirically validate our ideas and suggestions made in the previous sections. In particular, the goal of the experimental section is two-fold:

(1) to validate the argument that high auto-correlation through Direct features and medium/low auto-correlation through Indirect features is essential for CC to significantly outperform IC and
(2) to verify if the suggested baseline models serve as reasonable yardsticks in

evaluating sophisticated CC models.

A high level description of the tasks we set ourselves, to accomplish the above stated goals is given below. The details about classification models and experimental setups follow this high level description.

(1) *Task 1: To perform experiments that cover the full spectrum of possibilities identified by the level of auto-correlation through Direct and Indirect features.* We accomplish this by conducting experiments on synthetic data which span the space of all 9 possibilities and in addition give us control over the respective settings. The observations from these experiments concur with our predictions in Table I and Table II.

(2) *Task 2: To check if our predictions are consistent with the behavior seen in real world settings.* To accomplish this we conduct experiments on real datasets which cover some of the more interesting categories out of the possible 9 categories. In particular, we perform experiments on three datasets Cora [McCallum et al. 2000], Internet Movie Database (IMDb) [Neville and Jensen 2002] and UW-CSE [Richardson and Domingos 2006]. The Cora dataset resembles the case where the auto-correlation through Direct features is high and the auto-correlation through Indirect features is low/medium. The IMDb dataset we consider has no Direct features while the auto-correlation through Indirect features is reasonably high. Thus, this dataset resembles the case where the auto-correlation through Direct features is low and the auto-correlation through Indirect features is high. The UW-CSE dataset resembles the case where auto-correlation through both Direct and Indirect features is high.
The observations from these experiments strengthen our arguments regarding the predicted behavior.

In all our experiments we vary the proportion of known class labels from 0% to 30% to 90%, to observe its influence on the classification performance of the various models. To compare the performance of the CC and IC models we report their respective errors and verify if the observed difference in performance is statistical significant by computing $p-values$ of the two tailed paired t-test [Dietterich 1998]. The null hypothesis in this case is as follows;

H_0: The two models are equivalent.

Hence, smaller the $p-value$ the less likely it is that the null hypothesis is true and the models equivalent. As is generally the case, we reject H_0 if $p < 0.05$.

Classification Models: In the experiments we compare IC and DRN against two state-of-the-art collective classification models, i) Markov Logic Networks (MLN) [Richardson and Domingos 2006; Domingos and Richardson 2004] and ii) Relational Dependency Networks (RDN) [Neville and Jensen 2003]. The MLN is learned, a) generatively and b) discriminatively using the tool Alchemy [Kok et al. 2005]. For each type of learning we performed 2 types of inference 1) Maximum aposteriori (MAP) inference and 2) Markov Chain Monte Carlo (MCMC) inference (1000 runs). We observed that the results for discriminative and generative learning each with

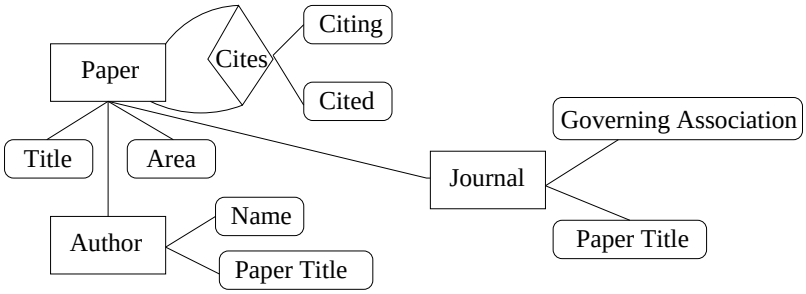

Fig. 6. Relational schema for synthetic experiments.

$XYL\downarrow, C\rightarrow$	C_1	C_2
$x_1y_1l_1$	p_1	p_2
$x_1y_1l_2$	p_3	p_4
$x_1y_1l_3$	p_5	p_6
$x_1y_2l_1$	p_7	p_8
$\vdots$	$\vdots$	$\vdots$
$x_4y_2l_3$	p_{47}	p_{48}

Table III. Above we see the structure of the synthetic data generator. For notational convenience we denote attributes Name, Governing Association and Area by X, Y and C respectively. The attribute L (i.e. (Citing,Cited)) specifies the connectivity in the graph. In particular, any 2 points having the same value of L are linked to each other. Each cell in the table contains the probability ($\sum_{q=1}^{48} p_q = 1$) of observing the specific input-output pair. The probabilities in the cells help in controling the level of auto-correlation through the different features. Their exact values, which are determined by the desired level of auto-correlation are given in the Appendix.

MAP and MCMC inference were qualitatively the same and hence we report results for just one combination of learning and inference i.e. for discriminative learning with MCMC inference. The conditional probability distributions (CPDs) in RDN are learned using Relational Bayes Classifiers (RBC) for the synthetic experiments (since there are few attributes) and Relational probability trees (RPT) [Neville 2006] for the experiments on real data (since they generally have better performance than RBC learning when number of features is large [Neville and Jensen 2007]). The inference is performed on the sample obtained after performing Gibbs sampling (burn-in is 100, number of samples is 1000) using the learned CPDs.

For the DRN model when class labels of none of the neighbors are known for a particular instance, we assign it a label based on the empirical class priors i.e. priors computed from the labels in the training set.

For IC we use different models depending on the relational dataset. This choice of model on real datasets is driven by the models that were shown to be successful in previously published work.

Setup for synthetic data experiments: The synthetic data is generated based on the relational schema in Figure 6. We use this schema, since it makes the synthetic setup simple enough to understand but complex enough to accommodate the necessary dependencies i.e. both Direct and Indirect features. It is important

to note here, that the necessary conditions we have provided are applicable *after* feature selection. Hence, choosing the appropriate aggregation function (viz. average(), exists() etc.) for an Indirect feature (viz. Age, Name etc.) giving rise to a new Indirect feature (viz. average(Age), exists(Name) etc.) is a preprocessing step prior to checking our conditions. Given this, we design the data generation model to have the right set of features with no further need for building new features through aggregation.

As we have mentioned before, we assign Area to be the class attribute which takes 2 values, AI and Graphics. We thus have instances belonging to either of 2 classes in the synthetic datasets that we generate. The attribute Name in *Author* takes 4 values and the attribute Governing Association in *Journal* takes 2 values (e.g. IEEE and ACM). The attribute Title (which is same as Paper Title in *Author* and *Journal*) in *Paper* is unique for each individual paper and hence does not help in determining the class label Area of the paper. We thus eliminate this attribute in the data generation models we consider. Generally, relational datasets have very few (if any) independent components through Direct features [5] in the data graph. Hence, we generate data that can have *utmost* 3 independent components. This is enforced by the attribute L in table III. Thus, the connectivity through Direct and Indirect features can be arbitrarily high.

With this, *Paper* has Indirect features Name, Governing Association (since Title and Paper Title have been purged) and the Direct feature (Citing,Cited). Consequently, auto-correlation through Direct features here implies auto-correlation through attributes of *Cites* and auto-correlation through Indirect features here implies auto-correlation through attributes of *Author* and *Journal*.

Training and Testing: We generate datasets of size 3000 and use 2000 samples for training and the remaining 1000 samples for testing respectively. We then use hold-out estimation to estimate the error of the classifiers trained on this data. We perform the above procedure 50 times and report the average error. As mentioned before, the equivalence in the performance of the respective algorithms is evaluated through hypothesis testing.

IC Model: For independent classification we form a contingency table from the training dataset like the one shown in Figure 5c, with the rows denoting author names (author1 and author2), columns the area of the papers (AI and Graphics) and the cells (intersection of rows and columns) containing the respective counts. The independent classifier we use in here, classifies a test instance into the class that is most numerous in the corresponding row of the contingency table. For example, a paper authored by author1 would be classified into AI, if the cell corresponding to author1 and AI has a larger count than the one corresponding to author1 and Graphics.

Setup for real data experiments: The experiments on real data are performed on 3 datasets namely; Cora [McCallum et al. 2000] – dataset of research papers,

[5]Unless they are absent. However, this would map to the low AD and high/medium/low AID case described in the paper

IMDb [Neville and Jensen 2002] – dataset of movies and UW-CSE [Richardson and Domingos 2006] – dataset containing information about the University of Washington Department of Computer Science and Engineering[6]. We now describe each of these datasets in detail and provide other relevant information regarding training and testing mechanisms, auto-correlation through Direct and Indirect features and the IC model used on each of these datasets.

(1) **Cora:** This dataset consists of research papers belonging to various fields in computer science. The papers we consider (similar to previous work [Macskassy and Provost 2003; Neville and Jensen 2007; Getoor and Taskar 2007]) belong to the field machine learning and the classification task is to classify each of the papers into the appropriate sub-fields (such as reinforcement learning etc.). The papers are linked to each other through citations, common authors, year of publication, common journals etc.

Training and Testing: We selected 4330 papers belonging to machine learning. From this set we sampled 1669 papers published between 1993 and 1998. We trained on papers from a particular year and tested on the papers belonging to the subsequent year. In the resulting data graphs all neighbors up to two links away were considered. Along with the citation information we considered seven attributes in the dataset namely; Topic (i.e. sub-field), Author rank (i.e. first author or second author), Month, Year, Paper type (e.g. tech report), Journal name-prefix (e.g. IEEE) and Book-role (e.g. conference). This setup is very similar to the one used in [Neville and Jensen 2007].

Auto-correlation: Considering the above description of the attributes and links that are chosen for training and testing, the dataset is known to exhibit high auto-correlation through Direct features (i.e. through citations) and low/medium through Indirect features (i.e. its own attributes and attributes of related types) [Neville and Jensen 2007].

IC Model: The independent classification model we choose in this case is a RPT. This model was compared against RDN on Cora in [Neville and Jensen 2007].

(2) **IMDb:** This dataset contains information about actors, directors, producers and studios associated with a movie. The classification task is to ascertain if the opening weekend box-office receipts of a movie is more than $2 million.

Training and Testing: We selected 1382 movies released in the US between 1996 and 2001. We trained on movies from a particular year and tested on the movies belonging to the subsequent year. In the resulting data graphs all neighbors up to two links were considered. We considered 8 attributes in the dataset namely; Receipts (the class label), First (the first movie by director or studio), Award (if an actor or director has won an Academy award), In-US (if

[6]All these datasets were obtained from http://alchemy.cs.washington.edu/

a studio is in the US), Genre (the movies type i.e. comedy, horror), Hsx-rating (actors value on the Hollywood Stock Exchange), Birth-year and Gender. This setup like in the case of Cora is very similar to the one used in [Neville and Jensen 2007].

Auto-correlation: Considering the above description of the attributes and links that are chosen for training and testing, the dataset is known to exhibit high auto-correlation through Indirect features (i.e. its own attributes and attributes of related types) while Direct features are not present [Neville and Jensen 2007]. Since, there are no linked input pairs through Direct features the auto-correlation through them is zero. This as mentioned before corresponds to the case wherein there is low auto-correlation through Direct features and high auto-correlation through Indirect features.

IC Model: The independent classification model we choose in this case is a RPT. This model was compared against RDN on IMDb in [Neville and Jensen 2007].

(3) **UW-CSE:** This dataset contains information about the UW computer science department. The dataset consists of people being either students or professors. The dataset has information regarding which course is taught by whom, who are the teaching assistants for a course, the publication record of a person, the phase in which a person is (i.e. pre-qualifier, post-qualifier), the position of a person (i.e. faculty, affiliate faculty etc.), years in a program and the adviser (or temporary adviser) of a student (given by "advisedby" links which are Direct features). The classification task is to find out if a person is a Student or Professor.

Training and Testing: The dataset has 442 people and is divided into five parts; ai.db, graphics.db, theory.db, language.db and systems.db. We performed 5-fold cross validation where we trained on the four parts and tested on the fifth. We used all the above mentioned features.

Auto-correlation: Considering the above description of the attributes and links that are chosen for training and testing, the dataset exhibits high auto-correlation through both Direct features (i.e. through advisedby links) and Indirect features (i.e. through phase of a person, teaching assistant for a course etc.). Here high auto-correlation through Direct features implies that the person advising is invariably a professor and the person being advised is generally a student unless the person advised also advises somebody else. The DRN in this case looks at the immediate neighbors and checks to see if any neighbor is being advised by the person in question. If the person does advise somebody, it classifies the person as professor else as student.

IC Model: The independent classification algorithm we choose in this case is C4.5 [Quinlan 1993]. In the papers we surveyed that used the UW-CSE

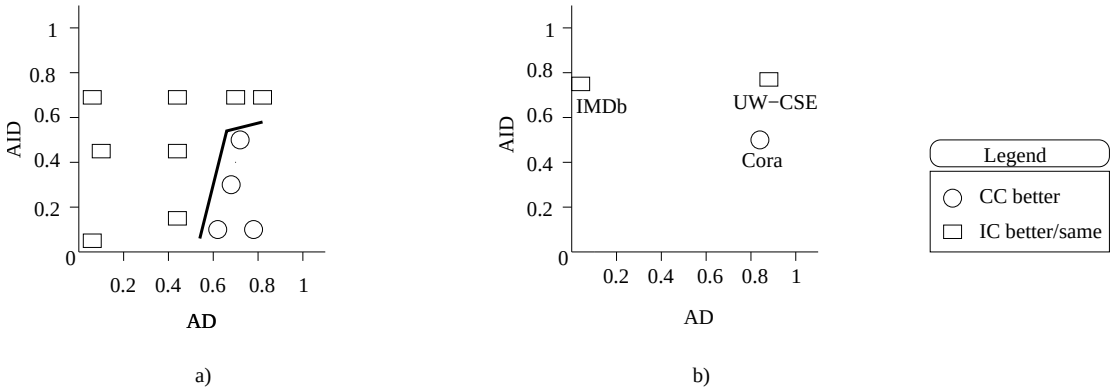

Fig. 7. Relative performance of CC vs IC for varying levels of AD and AID is shown in a). The dark reverse "L" shaped line depicts the consequent partitioning of the space where the region below the line represents AD and AID levels for which CC significantly outperforms IC. In b), we observe a similar trend for the real datasets that is consistent with a).

Model	MLN,IC			RDN,IC		
% labels known	0	30	90	0	30	90
AD,AID						
0.67,0.67 (HH)	0.05,0.04	0.04,0.04	0.04,0.04	0.06,0.04	0.04,0.04	0.04,0.04
0.81,0.68 (HH)	0.07,0.08	0.03,0.08	0.02,0.08	0.06,0.08	0.04,0.08	0.02,0.08
0.67,0.30 (HM)	**0.23,0.36**	**0.15,0.36**	**0.13,0.36**	**0.25,0.36**	**0.14,0.36**	**0.13,0.36**
0.71,0.50 (HM)	**0.13,0.24**	**0.08,0.24**	**0.09,0.24**	**0.12,0.24**	**0.07,0.24**	**0.08,0.24**
0.75,0.15 (HL)	**0.27,0.44**	**0.13,0.44**	**0.11,0.44**	**0.31,0.44**	**0.12,0.44**	**0.14,0.44**
0.64,0.15 (HL)	**0.34,0.45**	**0.31,0.45**	**0.32,0.45**	**0.33,0.45**	**0.29,0.45**	**0.30,0.45**
0.42,0.67 (MH)	0.09,0.05	0.08,0.05	0.09,0.05	0.06,0.05	0.05,0.05	0.07,0.05
0.45,0.45 (MM)	0.25,0.27	0.23,0.27	0.24,0.27	0.28,0.27	0.27,0.27	0.25,0.27
0.45,0.16 (ML)	0.36,0.4	0.33,0.4	0.34,0.4	0.38,0.4	0.35,0.4	0.34,0.4
0,0.67 (LH)	0.06,0.06	0.07,0.06	0.06,0.06	0.08,0.06	0.08,0.06	0.06,0.06
0.12,0.45 (LM)	0.25,0.26	0.27,0.26	0.25,0.26	0.27,0.26	0.26,0.26	0.24,0.26
0.06,0.06 (LL)	0.49,0.47	0.51,0.47	0.47,0.47	0.51,0.47	0.48,0.47	0.49,0.47
0.81,0.51 (Cora)	**0.15,0.45**	**0.1,0.45**	**0.08,0.45**	**0.13,0.45**	**0.09,0.45**	**0.09,0.45**
0,0.76 (IMDb)	0.17,0.22	0.15,0.22	0.14,0.22	0.15,0.22	0.13,0.22	0.11,0.22
0.89,0.78 (UW-CSE)	0.15,0.1	0.09,0.1	0.09,0.1	0.2,0.1	0.09,0.1	0.08,0.1

Table IV. Errors of the classification models with varying levels of auto-correlation through Direct and Indirect features (leftmost column) and with varying percentage of known class labels is shown above. The entries in bold indicate that the difference in performance of the two algorithms was statistically significant.

dataset [Richardson and Domingos 2006; Kok and Domingos 2005; Singla and Domingos 2005; Mihalkova and Mooney 2007; Mihalkova et al. 2007], the task was never to compare IC and CC (rather link prediction or entity resolution etc.) and hence we choose a standard IC algorithm namely; C4.5 for this case.

Model	DRN,IC		
% labels known	0	30	90
AD,AID			
0.67,0.67 (HH)	0.05,0.04	0.05,0.04	0.05,0.04
0.81,0.68 (HH)	**0.41,0.08**	0.04,0.08	0.05,0.08
0.67,0.30 (HM)	0.31,0.36	**0.17,0.36**	**0.15,0.36**
0.71,0.50 (HM)	0.21,0.24	**0.09,0.24**	**0.07,0.24**
0.75,0.15 (HL)	0.42,0.44	**0.19,0.44**	**0.18,0.44**
0.64,0.15 (HL)	0.41,0.45	**0.35,0.45**	**0.32,0.45**
0.42,0.67 (MH)	**0.3,0.05**	**0.27,0.05**	**0.26,0.05**
0.45,0.45 (MM)	0.31,0.27	0.29,0.27	0.29,0.27
0.45,0.16 (ML)	0.41,0.4	0.32,0.4	0.33,0.4
0,0.67 (LH)	**0.47,0.06**	**0.51,0.06**	**0.48,0.06**
0.12,0.45 (LM)	**0.41,0.26**	**0.38,0.26**	**0.39,0.26**
0.06,0.06 (LL)	0.51,0.47	0.48,0.47	0.48,0.47
0.81,0.51 (Cora)	0.5,0.45	**0.35,0.45**	**0.2,0.45**
0,0.76 (IMDb)	NA		
0.89,0.78 (UW-CSE)	0.15,0.1	0.12,0.1	0.11,0.1

Table V. Errors of the classification models with varying levels of auto-correlation through Direct and Indirect features (leftmost column) and with varying percentage of known class labels is shown above. The entries in bold indicate that the difference in performance of the two algorithms was statistically significant.

Notation: AD and AID represent the level of auto-correlation through Direct and Indirect features respectively. HH denotes high AD and high AID, HM denotes high AD and medium AID, HL denotes high AD and low AID, MH denotes medium AD and high AID, MM denotes medium AD and medium AID, ML denotes medium AD and low AID, LH denotes low AD and high AID, LM denotes low AD and medium AID and LL denotes low AD and low AID.

5.1 Observations

Validation of necessary conditions: In Figure 7 and table IV we observe that CC significantly outperforms IC when auto-correlation through Direct features is high and auto-correlation through Indirect features is low/medium. The dark line in figure 7a partitions the AD/AID space, where the region below the line indicates the levels of auto-correlation needed for CC to be superior to IC. Knowledge of a fraction of the class labels helps to improve the performance of CC in these cases, however, even when no labels are known CC is a more desirable alternative.

Evaluation of baseline classifiers: From table V we see that DRN is significantly superior to IC only when auto-correlation through Direct features is high, auto-correlation through Indirect features is low/medium and some of the class labels are known with certainty. This trend is seen for synthetic as well as real data. Given these observations and the observations in table IV, IC seems to be a reasonable baseline for the other cases as suggested in table II.

5.2 Overview

From the above experimental results we see that for CC to significantly outperform IC the dataset should exhibit either HM or HL. Since, one of the major goals of

this paper was to decipher stronger necessary conditions for this to occur and not sufficient conditions, the synthetic datasets generated exhibited high linkage which is known to favor CC techniques. Hence, the evidence provided in this section makes a strong case in support of the claim that CC cannot outperform IC unless the dataset exhibits HM or HL (and not vice-versa). Moreover, the experiments on real data are consistent with these findings, further strengthening our claim.

6. DISCUSSION

In the previous section we empirically validated our claims. In this section we look at transformations of a given relational data graph which have been considered in the literature [Kok and Domingos 2007; Macskassy and Provost 2003; 2007; Gallagher et al. 2008] and discuss its effects on the claims made in this paper. We also discuss the implications of the observations reported in [Jensen et al. 2004], where Indirect features were increased, has on our analysis.

Data graph transformations: Let us first consider the scenario where Direct features are replaced by Indirect features. A simple way of accomplishing this is by introducing an extra object. The number of attributes in the object is equal to the number of different Direct features. For a particular Direct feature the corresponding attribute in this new object has the same value for objects to be classified that are linked by that Direct feature and different values otherwise. After introducing such an object the Direct features are dropped and hence the relational dataset just has Indirect features. For example, in Figure 4 *Cites* may be replaced by an object with one attribute which has the same value for papers that are related through *Cites* and different values otherwise. Such a transformation however, would increase the dimensionality of the space making learning more expensive and potentially increasing variance of the learned classifier. This process of introducing new objects (or attributes) and analyzing their effects is a research problem in itself. Considering the added complexity of modeling (i.e. introducing an appropriate object/objects with attributes) and implementation it is indeed not an ideal choice as a simple baseline. An IC model trained over this transformed dataset would probably perform much better compared to its performance on the original dataset only if the paths through the added Indirect features exhibit high auto-correlation that was missing in the original paths (through Indirect features). This would be possible only if the auto-correlation through the replaced Direct features was high. Hence, the performance of IC may become comparable to the original collective classifier, though not necessarily. In either case the claims made in the paper are still valid since the conditions provided are necessary conditions for CC to outperform IC and not sufficient conditions.

Let us now consider the scenario where Indirect features are replaced by Direct features. This is seen in [Macskassy and Provost 2003; 2007] where paths through Indirect features (of length 2) are replaced by Direct features. One of the main challenges in performing this transformation is choosing the appropriate set of interactions (called edge selection in [Macskassy and Provost 2007]) that enhance classification accuracy. Certain heuristics are discussed in [Macskassy and Provost 2007] to accomplish this but it is still an important open problem. Moreover, the

transformed data graph which resembles a network of homogeneous nodes requires that a fraction of the labels be known and that the known labels be roughly uniformly distributed in the graph for the RN approaches to work well. This requirement is important since if labels of only a localized portion of the graph are known then the error of a nearest neighbor like approach would most likely be high on the remaining graph. Though the requirement is important it is a strong requirement in practice. For example, in a company dataset where we want to classify employees based on their performance, a manager may know about employees in his branch but not about many in other branches of the same firm. Given these difficulties it would probably be more feasible in many cases to have IC models as baselines when the auto-correlation through Indirect features is high as opposed to transforming the graph and using CC models. Assuming that we select the appropriate set of interactions and assuming that we know the right set of labels, the nearest neighbor like CC models can become competitive with IC models in the transformed graph, given that auto-correlation through Indirect features is high (and low through Direct features) in the original graph. However, since the objects that are linked through Direct features in the new graph were previously linked through Indirect features in the original graph, a nearest neighbor approach in both domains should be comparable. Moreover, there are a wide range of sophisticated algorithms such as SVMs, Neural networks, Decision trees, etc. that can be readily applied to the original graph but not to the new graph. Hence, our claims in the paper still hold.

Behavior with increasing Indirect features: In [Jensen et al. 2004] the authors observe that with increasing number of attributes (Indirect features) the performance of a nearest neighbor collective classification model based only on Direct features (denoted by C1 in that paper) improves relative to other collective (denoted by RC1) and independent classification models (denoted by R1) that use those attributes. Thus the C1 models defined in that paper are a special case of DRN, the RC1 models are similar to MLN and RDN (i.e. these models use Direct and Indirect features) and the R1 model is an IC model. The dataset they run their experiments on is auto-correlated through Direct features and certain Indirect features. An explanation for the observed behavior given in that paper is that the added attributes are not particularly discriminative and hence only end up increasing the variance of the trained classifiers (due to increased state space) without reducing the bias significantly. Based on this explanation, the scenario where attributes are added and the data is auto-correlated through Direct features maps to the case of high AD and medium/low AID. Hence, a DRN (or C1) turns out to be a good choice in such a setting.

7. FUTURE WORK

In this work, we have provided conditions that are essential for CC to outperform IC based on arguments that focus on the amount of predictive information that is available through the 2 types of features. It is important to stress here that these conditions are necessary and not sufficient. In other words, the statement this paper makes is that CC cannot outperform IC unless the dataset falls under one of the two categories namely; HL or HM. However, even if the dataset belongs

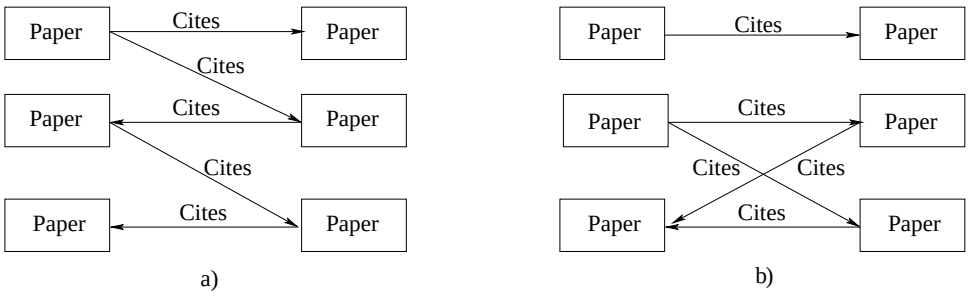

Fig. 8. Two data graphs with the same number of links and the same sample size are shown above. However, the link distribution/connectivity is different in the 2 data graphs, which may affect sufficient conditions for CC to outperform IC.

to HL or HM, the paper does not claim that CC will most definitely outperform IC by a significant margin. Making such a claim would imply that the conditions are sufficient.

In the future, it would be interesting to decipher conditions that are sufficient for the same, if we consider that any classification algorithm that models the information in Direct features (be it through graph transformations) is a CC technique, though the converse may not be true. In addition to observations made in this paper, we believe that this would require delving into the properties related to linkage (through Direct features) of the graph; something we have not thoroughly explored in this work. Studying linkage would involve studying the effects that properties such as number of links in the graph, the link distribution i.e. how the links are spread throughout the graph, would have on the relative performance of CC and IC. For example, CC might perform superior to IC on a data graph in figure 8a but not in figure 8b, eventhough the number of links and the auto-correlation are the same. To this end, we would probably have to build an estimator for linkage that depends on these two properties and that captures its effects on the performance of CC and IC in a consistent manner. By consistent we mean that we can set a threshold, say α for this estimator and be able to make statements such as, if the auto-correlation through Direct and Indirect features lies in a specific range with the linkage greater than α then these conditions are *sufficient* for CC to outperform IC. A possible estimator of linkage could be $\frac{N-k}{N-1}$ where N is the dataset size and k is the number of independent components in the graph. Such an estimator would lie in the interval [0,1] with 0 implying the data points are independent and 1 implying all data points linked through one or multiple hops. The estimator would vary with the link distribution eventhough, the number of links might be identical in two different data graphs, thus capturing our intuitions. However, careful investigation needs to be done to test the validity of such ideas in the future.

8. CONCLUSION

In this paper we pinpointed the necessary conditions under which collective classification should be preferred over independent classification. In view of this, we split the feature space into Direct and Indirect features with auto-correlation through

these features being either high, medium or low leading to 9 possible cases. We showed that collective classification is preferable to independent classification for 2 of these cases, that is when i) auto-correlation through Direct features is high and ii) auto-correlation through Indirect features is low or medium. In the remaining 7 cases independent classification was more than acceptable. We also suggested baseline models that can be used to evaluate state-of-the-art collective classification models for each of these 9 cases. We introduced the Direct Relational Neighbor model (which classifies based on Direct features) as baseline for the above 2 cases wherein collective classification significantly outperformed independent classification and for the remaining 7 cases we recommended using an independent classification algorithm that is known to perform well for the particular application. In summary, by studying the similarities and differences in behavior between the two classification paradigms namely; collective classification and independent classification, we have tried to improve our current state of understanding as to when and why these two classification paradigms might be useful.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0448264.

REFERENCES

Angin, P. and Neville, J. 2008. A shrinkage approach for modeling non-stationary relational autocorrelation. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 707–712.

Chakrabarti, S., Dom, B., and Indyk, P. 1998. Enhanced hypertext categorization using hyperlinks. In *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, L. M. Haas and A. Tiwary, Eds. ACM Press, New York, US, Seattle, US, 307–318.

Dietterich, T. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation 10*, 1895–1923.

Domingos, P. and Richardson, M. 2004. Markov logic: A unifying framework for statistical relational learning. In *Proceedings of the ICML 2004 Workshop on Statistical Relational Learning and its Connections to Other Fields*. 49–54.

Gallagher, B., Tong, H., Eliassi-Rad, T., and Faloutsos, C. 2008. Using ghost edges for classification in sparsely labeled networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, 256–264.

Getoor, L., Koller, D., and Small, P. 2004. Understanding tuberculosis epidemiology using probabilistic relational models. *Journal of Artificial Intelligence in Medicine 30*, 233–256.

Getoor, L. and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Jensen, D., Neville, J., and Gallagher, B. KDD 2004. Why collective inference improves relational classification.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

Kok, S. and Domingos, P. 2005. Learning the structure of markov logic networks. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*. ACM, New York, NY, USA, 441–448.

Kok, S. and Domingos, P. 2007. Statistical predicate invention. In *ICML '07: Proceedings of the 24th international conference on Machine learning*. ACM, New York, NY, USA, 433–440.

Kok, S., Singla, P., Richardson, M., and Domingos, P. 2005. The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, UW, http://www.cs.washington.edu/ai/alchemy/.

Macskassy, A. and Provost, F. KDD 2003. A simple relational classifier.

Macskassy, S. and Provost, F. 2007. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res. 8*, 935–983.

Malvestuto, F. 1989. A universal table model for categorical databases. *Inf. Sci. 49,* 1-3, 203–223.

McCallum, A., Nigam, K., Rennie, J., and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval 3,* 2, 127–163.

Mihalkova, L., Huynh, T., and Mooney, R. J. 2007. Mapping and revising markov logic networks for transfer learning. In *AAAI*. 608–614.

Mihalkova, L. and Mooney, R. 2007. Bottom-up learning of markov logic network structure. In *ICML '07: Proceedings of the 24th international conference on Machine learning*. ACM, New York, NY, USA, 625–632.

Neville, J. 2006. Statistical models and analysis techniques for learning in relational data. Ph.D. Thesis, University of Massachusetts Amhers.

Neville, J. and Jensen, D. 2002. Data mining in social networks. In *In National Academy of Sciences Symposium on Dynamic Social Network Analysis*.

Neville, J. and Jensen, D. 2005. Leveraging relational autocorrelation with latent group models. In *MRDM '05: Proceedings of the 4th international workshop on Multi-relational mining*. ACM, New York, NY, USA, 49–55.

Neville, J. and Jensen, D. 2007. Relational dependency networks. *J. Mach. Learn. Res. 8*, 653–692.

Neville, J. and Jensen, D. AAAI 2000. Iterative classification in relational data.

Neville, J. and Jensen, D. KDD 2003. Collective classification with relational dependency networks.

Perlich, C. and Provost, F. 2006. Distribution-based aggregation for relational learning with identifier attributes. *Mach. Learn. 62,* 1-2, 65–105.

Quinlan, R. 1993. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann.

Richardson, M. and Domingos, P. 2006. Markov logic networks. *Mach. Learn. 62,* 1-2, 107–136.

Sen, P., Namata, G. M., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. 2008. Collective classification in network data. *AI Magazine 29,* 3.

Singla, P. and Domingos, P. 2005. Discriminative training of markov logic networks. In *AAAI* (2005-09-01), M. M. Veloso and S. Kambhampati, Eds. AAAI Press / The MIT Press, 868–873.

Taskar, B., Abbeel, P., and Koller, D. 2002. Discriminative probabilistic models for relational data. In *In Proc. 18th Conference on Uncertainty in AI*. 485–492.

Taskar, B., Segal, E., and Koller, D. 2001. Probabilistic classification and clustering in relational data. In *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, B. Nebel, Ed. Seattle, US, 870–878.

Vapnik, V. 1998. *Statistical Learning Theory*. Wiley & Sons.

9. APPENDIX

The cell probabilities for the synthetic generation process are given below. The auto-correlation is computed using Pearson's contingency coefficient with respect to the null hypothesis that all cells have equal counts. We use this hypothesis since it directly relates to how predictive the various inputs are likely to be. Note that "-do-" indicates that the same values are repeated for the respective columns.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$	$x_4y_2l_3$
C_1	0.04	0.04	-do-	0.04
C_2	0.0017	0.0017	-do-	0.0017

Table VI. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.67, AID = 0.67.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$									$x_2y_2l_3$
C_1	0.1	0	0	0	0	0	0.14	0	0	0.005	0	0
C_2	0	0.005	0	0	0.1	0.085	0	0.005	0.04	0	0.1	0
XYL	$x_3y_1l_1$	$x_3y_1l_2$	$\cdots$									$x_4y_2l_3$
C_1	0.1	0	0	0.005	0	0	0.1	0	0	0.005	0	0
C_2	0	0.005	0	0	0.1	0	0	0.005	0	0	0.1	0

Table VII. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.81, AID = 0.68.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$									$x_2y_2l_3$
C_1	0.04	0.04	0.04	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.04	0.04	0.04
C_2	0.0017	0.0017	0.0017	0.04	0.04	0.04	0.04	0.04	0.04	0.0017	0.0017	0.0017
XYL	$x_3y_1l_1$	$x_3y_1l_2$	$\cdots$									$x_4y_2l_3$
C_1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
C_2	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017

Table VIII. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.42, AID = 0.67.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$									$x_2y_2l_3$
C_1	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
C_2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
XYL	$x_3y_1l_1$	$x_3y_1l_2$	$\cdots$									$x_4y_2l_3$
C_1	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
C_2	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017

Table IX. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0, AID = 0.67.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	...									$x_2y_2l_3$
C_1	0.0017	0.0017	0.04	0.0017	0.0017	0.04	0.0017	0.0017	0.04	0.0017	0.0017	0.04
C_2	0.04	0.04	0.0017	0.04	0.04	0.0017	0.04	0.04	0.0017	0.04	0.04	0.0017
XYL	$x_3y_1l_1$	$x_3y_1l_2$	...									$x_4y_2l_3$
C_1	0.0017	0.0017	0.04	0.0017	0.0017	0.04	0.0017	0.0017	0.04	0.0017	0.0017	0.04
C_2	0.04	0.04	0.0017	0.04	0.04	0.0017	0.04	0.04	0.0017	0.04	0.04	0.0017

Table X. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.67, AID = 0.3.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	...									$x_2y_2l_3$
C_1	0.056	0.0417	0.0006	0.056	0.0417	0.0006	0.056	0.0417	0.0006	0.056	0.0417	0.0006
C_2	0.0016	0.005	0.02	0.0016	0.005	0.02	0.0016	0.005	0.02	0.0016	0.005	0.02
XYL	$x_3y_1l_1$	$x_3y_1l_2$	...									$x_4y_2l_3$
C_1	0.056	0.0417	0.0006	0.056	0.0417	0.0006	0.056	0.0417	0.0006	0.056	0.0417	0.0006
C_2	0.0016	0.005	0.02	0.0016	0.005	0.02	0.0016	0.005	0.02	0.0016	0.005	0.02

Table XI. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.71, AID = 0.5.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	...	$x_4y_2l_3$
C_1	0.0313	0.0313	-do-	0.0313
C_2	0.0104	0.0104	-do-	0.0104

Table XII. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.45, AID = 0.45.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	...									$x_2y_2l_3$
C_1	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0313	0.0313	0.0313
C_2	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0104	0.0104	0.0104
XYL	$x_3y_1l_1$	$x_3y_1l_2$	...									$x_4y_2l_3$
C_1	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313
C_2	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104	0.0104

Table XIII. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.12, AID = 0.45.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	...									$x_2y_2l_3$
C_1	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017
C_2	0.0037	0.058	0.01	0.0037	0.058	0.01	0.0037	0.058	0.01	0.0037	0.058	0.01
XYL	$x_3y_1l_1$	$x_3y_1l_2$	...									$x_4y_2l_3$
C_1	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017
C_2	0.0037	0.058	0.01	0.0037	0.058	0.01	0.0037	0.058	0.01	0.0037	0.058	0.01

Table XIV. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.75, AID = 0.15.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$									$x_2y_2l_3$
C_1	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017
C_2	0.0317	0.03	0.01	0.0317	0.03	0.01	0.0317	0.03	0.01	0.0317	0.03	0.01
XYL	$x_3y_1l_1$	$x_3y_1l_2$	$\cdots$									$x_4y_2l_3$
C_1	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017	0.05	0.0017	0.0017
C_2	0.0317	0.03	0.01	0.0317	0.03	0.01	0.0317	0.03	0.01	0.0317	0.03	0.01

Table XV. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.64, AID = 0.15.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$									$x_2y_2l_3$
C_1	0.0104	0.0313	0.0313	0.0104	0.0313	0.0313	0.0104	0.0313	0.0313	0.0104	0.0313	0.0313
C_2	0.0313	0.0104	0.0104	0.0313	0.0104	0.0104	0.0313	0.0104	0.0104	0.0313	0.0104	0.0104
XYL	$x_3y_1l_1$	$x_3y_1l_2$	$\cdots$									$x_4y_2l_3$
C_1	0.0104	0.0313	0.0313	0.0104	0.0313	0.0313	0.0104	0.0313	0.0313	0.0104	0.0313	0.0313
C_2	0.0313	0.0104	0.0104	0.0313	0.0104	0.0104	0.0313	0.0104	0.0104	0.0313	0.0104	0.0104

Table XVI. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.45, AID = 0.16.

XYL	$x_1y_1l_1$	$x_1y_1l_2$	$\cdots$	$x_4y_2l_3$
C_1	0.0222	0.0222	-do-	0.0222
C_2	0.0196	0.0196	-do-	0.0196

Table XVII. The auto-correlations observed by sampling $N = 3000$ samples from this distribution are, AD = 0.06, AID = 0.06.