Learning Bayesian Network Using Parse Trees for Extraction of Protein-Protein Interaction

Pedro Nelson Shiguihara-Juárez and Alneu de Andrade Lopes

Institute of Mathematical Sciences and Computation
University of São Paulo
{pshiju,alneu}@icmc.usp.br

Abstract. Extraction of protein-protein interactions from scientific papers is a relevant task in the biomedical field. Machine learning-based methods such as kernel-based represent the state-of-the-art in this task. Many efforts have focused on obtaining new types of kernels in order to employ syntactic information, such as parse trees, to extract interactions from sentences. These methods have reached the best performances on this task. Nevertheless, parse trees were not exploited by other machine learning-based methods such as Bayesian networks. The advantage of using Bayesian networks is that we can exploit the structure of the parse trees to learn the Bayesian network structure, i.e., the parse trees provide the random variables and also possible relations among them. Here we use syntactic relation as a causal dependence between variables. Hence, our proposed method learns a Bayesian network from parse trees. The evaluation was carried out over five protein-protein interaction benchmark corpora. Results show that our method is competitive in comparison with state-of-the-art methods.

1 Introduction

The automation of protein-protein interaction (PPI) extraction from scientific papers is a critical and relevant task in the biomedical field. PPIs are important to understand the cell behavior and, consequently, to develop new drugs. Initially, manual extractions (curations) were used to perform this task. However, the PPI extraction has been adversely limited by the growing amount of papers and the time-consuming task involved [1]. Although this task has been addressed by various computational approaches, the extraction of PPI still challenges the Machine Learning community. For example, a sentence containing names of several proteins could involve multiple interactions. In addition, there is a large number of possibilities to express the same idea utilizing natural language in written form. Since these problems affect the performance of computational approaches, the task is commonly considered as a binary classification problem in order to reduce its complexity [2]. It consists in detecting whether an interaction involved between a pair of protein names exists or not.

Figure 1 illustrates the extraction of PPIs from a sentence[1] as a binary problem. Three pairs of protein names are candidates: *Actin−Iota toxin*, *Iota*

[1] This sentence belongs to the Bioinfer corpus. ID: "BioInfer.d27.s0".

A. Gelbukh (Ed.): CICLing 2013, Part II, LNCS 7817, pp. 347–358, 2013.

toxin−Profilin and *Actin−Iota toxin*, and the output for these pairs of protein names are "true", "false" and "true" respectively. Thus, only two pairs of protein names are considered interaction relationships.

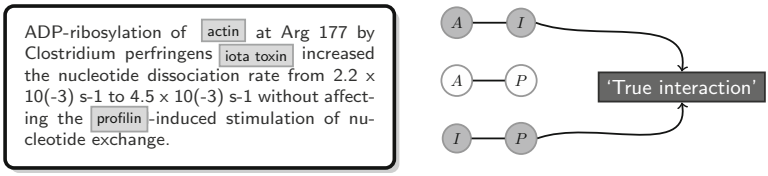

Fig. 1. *actin* (A), *iota toxin* (I) and *profilin* (P) proteins are evaluated in order to determine which pairs of these proteins describe an interaction. Two interactions were founded. The first one belongs to the *Actin−Iota toxin* (A,I) pair and, the last one belongs to the *Iota toxin−Profilin* (I,P) pair.

As stated earlier, the PPI task has been treated by different machine learning approaches, particularly, the kernel approaches using parse trees (PTs) [3] or dependency trees [2]. In contrast, there is a lack of recent research on Bayesian network (BN) models. Although BN models, using lexical features, have been proposed to solve the PPI problem before [1, 4], the PTs may play an important role to provide syntactic features for BN models. In BN, features and their relationships are modeled as a graph from unstructured data. However, this modeling may be enriched when relationships among features are also given from the training data set. In that sense, a PT represents syntactic information from a sentence in a tree form, where relationships are already given. Taking this into account, BNs can be learned from PTs. The rationale behind this hypothesis is that syntactic patterns among features may convey relevant information about the existence or not of PPIs.

To the best of our knowledge, this is the first work that proposes to learn a Bayesian network directly from parse trees to PPI extraction. The method combines the PTs, from sentences in training data, in order to create a (probably cyclic) graph. Then, the method removes edges and obtains a directed acyclic graph. Finally, we limit the maximum number of parents for each node in order to reduce the number of parameters and add a class node containing the values 'true interaction' or 'no interaction' in the BN. To complete the BN model (structure and parameters), the method uses the maximum likelihood estimation (MLE) [5] to calculate the parameters.

Models obtained by the proposed method were evaluated using five well-known PPI corpora. They are IEPA, AIMed, BioInfer, HPRD50 and LLL. Results demonstrated that the performance of our BN models are competitive compared to kernel-based methods applied on large corpora. However the performance of our models decrease when employed on small corpora. This limitation can be less latent since annotated corpora on PPI are becoming larger.

The reminder of the paper is organized as follows. In Section 2, we describe the related work on the PPI extraction problem. In Section 3, we explain the proposed method. In Section 4, we provide experimental results that demonstrate the competitive performance of the proposed method in comparison with the state-of-the-art methods. Finally, we conclude this work.

2 Related Work

According to [6], computational methods for PPI extraction can be organized in three groups. The first group consists of co-occurrence and rule-based approaches. The co-occurrence-based approaches evaluate the likelihood of two protein names co-occur in a same sentence. The rule-based approaches define a set of rules which is commonly applied on syntactic features, such as dependency trees [7]. These rules represent evidence of the existence of interactions. However, approaches based on co-occurrence lead to a low precision, whereas rule-based approaches lead to a low recall on the PPI extraction [1]. As a consequence, these methods have not succeeded on this task, needing further improvements. The second group is machine learning-based approaches which are commonly combined with natural language processing techniques. Recently, ensemble systems demonstrated a high performance in BioCreative II.5 [8]. Also, kernel-based machine learning approaches have been widely used in the PPI extraction task [2, 9–11, 6, 3, 12] with a high performance. They are considered state-of-the-art with the use of syntactic features. The third group is the combination from the approaches of the earlier two groups [7].

In machine learning-based methods, Dynamic Bayesian networks [4] were employed to solve the PPI extraction task in a multi-class context. In this case, words from sentences are used as features. However, the PPI corpora are getting larger and the use of all words as features in a Bayesian network increases computational cost. Bayesian networks learned by using Hill-climbing search algorithm [1] were also used in PPI extraction. These BN models use a fixed number of features (7 features). This small number of features could be little representative for all possible interactions existing in a sentence. A study [9] of the performance of the state-of-the-art methods in the PPI extraction indicates that these methods achieve between 19% and 30% of performance in terms of F-measure. In that sense, this study [9] proposes the use of an unified format of five benchmark corpora for more reliable evaluations. At present, these corpora are used by several works on PPI extraction problem and are also used in this work.

Currently, the kernel-based approaches are dominant on this task. Dependency trees were used to construct knowledge in a graph representation [2]. Afterwards, features are extracted from the graph. In the next step, these features are used on a regularized least squares kernel-based approach. In a similar way, kernel based on dependency paths [13] were used to cover different syntactic substructures and to obtain similarities (or dissimilarities) among sentences. Also, support vector machines using lexical and syntactic features [12] were used,

obtaining little improvements. In an effort to achieve better results, multiple kernels combining lexical and syntactic features [10] obtained a high performance in comparison with the state-of-the-art approaches. Factors to improve the performance, such as pruning parse trees and tuning parameters of support vector machines, were employed in a simple but effective kernel-based method [3]. As suggested in [3], the pruning factor can be used independently of the machine learning-based approach employed. Thus, we also use prune method for parse trees to improve the performance and to reduce irrelevant information. In BioCreative II.5 challenge, the OntoGene text mining environment [14] obtained the best results. The OntoGene system employed dependency trees, demonstrating the importance of syntactic features on the PPI extraction task.

3 Extraction of PPI Using a Probabilistic Model

In PPI extraction, the relation extraction typically consider binary relations. This means that we have $\binom{n}{2}$ instances from a sentence, where n is the number of proteins. For example, we have $\binom{3}{2} = 3$ instances from the sentence described in Figure 1, since there are 3 protein names. These instances named (A, I), (A, P) and (I, P), must be evaluated to identify if there exist an interaction or not. Thus, currently, the relation extraction task can be seen as a binary classification problem [2]. The goal of this classification problem is to calculate the maximum a posteriori ($\hat{y}$) of the random class variable ($\mathcal{C}$), and a pair of proteins is extracted if there exist an interaction ($\mathcal{C} = 1$). In both cases ($\mathcal{C} = 1$ and $\mathcal{C} = 0$), we are given a classifier model $\mathcal{M}$, a training data set $\mathcal{D}$ and a test instance x denoted by (prot1, prot2), since it is assumed a binary relation. The maximum a posteriori is obtained among values of the class $\mathcal{C}$ in equation (1).

$$\hat{y} = \arg_{\mathcal{C}} \max \{P(\mathcal{C} = 0 \mid \mathcal{M}, \mathcal{D}, x), P(\mathcal{C} = 1 \mid \mathcal{M}, \mathcal{D}, x)\} \tag{1}$$

Thus, we use a function for relation extraction according to equation (2).

$$extractRelation\,(\hat{y}, x) = \begin{cases} \hat{y} = 1 & interactsWith(x.prot1, x.prot2) \\ \hat{y} = 0 & \emptyset \end{cases} \tag{2}$$

where the relation $interactsWith(x.prot1, x.prot2)$ is extracted whether a $\mathcal{M}$ classifier predicts an interaction ($\hat{y} = 1$) for the pair of protein names corresponding to the test instance x. For all test cases, the relation $interactsWith$ is defined as 'true interaction'. In this work, parse trees are considered as instances in the training data set $\mathcal{D}$ and, $\mathcal{M}$ is our Bayesian network model induced from $\mathcal{D}$.

3.1 Learning a Bayesian Network from Parse Trees

Given a Bayesian network model $\mathcal{M} = \langle G_{\mathcal{M}}, \theta \rangle$, where $G_{\mathcal{M}}$ is its structure (i.e., a directed acyclic graph) and θ is the set of parameters of the BN, the goal is to learn the model from parse trees in $\mathcal{D}$. The graph $G_{\mathcal{M}} = \langle Z_{\mathcal{M}}, E_{\mathcal{M}} \rangle$ contains a set of nodes ($Z_{\mathcal{M}}$) and edges ($E_{\mathcal{M}}$).

The basic assumption of the method is that non-terminal and pre-terminal nodes in parse trees can be considered as random variables ($Z_{\mathcal{M}}$) in order to learn a Bayesian network. In addition, an edge between a parent node and its child node, in the parse trees, can be regarded as a directed edge ($E_{\mathcal{M}}$) in the Bayesian network structure. This information is used to infer how these potential random variables are related to each other. Thus, we explain how our method allows to learn the structure and the parameters of a Bayesian network model.

Bayesian Network Structure. Our method follows 4 steps to learn the network structure: (1) create a (probably cyclic) graph structure using the non-terminal and pre-terminal nodes from parse trees, (2) remove all the cycles in the graph structure, (3) limit the maximum number of parents to d for each node z_k using mutual information to rank and keep only the d *best parents*, and (4) add a bi-valued class node to the graph in order to infer whether a sentence has or not an interaction.

In the first step, we consider a training data set containing M parse trees. Each parse tree PT_i ($1 \leq i \leq M$) is defined as $PT_i = \langle Z_{PT_i}, E_{PT_i} \rangle$, where a non-terminal or a pre-terminal node $z_k \in Z_{PT_i}$ could also exist in other parse trees. In the i-th parse tree, PT_i, Z_{PT_i} and E_{PT_i} is the set of nodes and edges respectively. We aim to find the random variables of the Bayesian network $Z_{\mathcal{M}} = \{Z_{PT_1} \bigcup \ldots \bigcup Z_{PT_n}\}$. In this context, we considered the non-terminal and pre-terminal nodes from parse trees as random variables in the $Z_{\mathcal{M}}$ set. In a same way, the set of edges can be defined by $E_{\mathcal{M}} = \{E_{PT_1} \bigcup \ldots \bigcup E_{PT_n}\}$. An edge $e_x \in E_{\mathcal{M}}$ describes a relationship *parent* $\rightarrow$ *child* over the nodes in Z_{PT_i}. For example, assuming that we can learn a Bayesian network structure from only four parsed sentences (see Figure 2), the result $G'_{\mathcal{M}} = \langle Z'_{\mathcal{M}}, E'_{\mathcal{M}} \rangle$, after removing cycles, is shown in Figure 3.(a).

The second step involves the elimination of the cycles. We employ a depth-first search technique and put repeated nodes (with the corresponding parent) in a queue data structure. Since the preservation of the original relationships is desired, we only remove one edge in the queue. This edge contains the pair of nodes (a repeated node with its parent node) with the maximum distance from each other, when we apply the depth-first search. We do this second step again until no further repeated nodes are detected.

In the third step, each node $z_k \in Z_{\mathcal{M}}$ is restricted to have at most d number of parents. Such restriction minimizes the complexity of the Bayesian network structure in terms of the number of parameters. To select edges to be removed, we rank each edge containing a relationship between the node z_k and each of its parents $z_{\pi_{kj}} \in Z_{\mathcal{M}}$ by using the *mutual information* ($\mathcal{I}$) measure in equation (3). The node $z_{\pi_{kj}}$ is the j-th parent of the z_k node. The mutual information, between two variables z_k and $z_{\pi_{kj}}$, determines how similar is the joint distribution $p\left(z_k, z_{\pi_{kj}}\right)$ in comparison to the factored distribution $p\left(z_k\right) p\left(z_{\pi_{kj}}\right)$.

$$\mathcal{I} = \sum_{z_k} \sum_{z_{\pi_{kj}}} p(z_k, z_{\pi_{kj}}) \log \frac{p\left(z_k, z_{\pi_{kj}}\right)}{p\left(z_k\right) p\left(z_{\pi_{kj}}\right)} \tag{3}$$

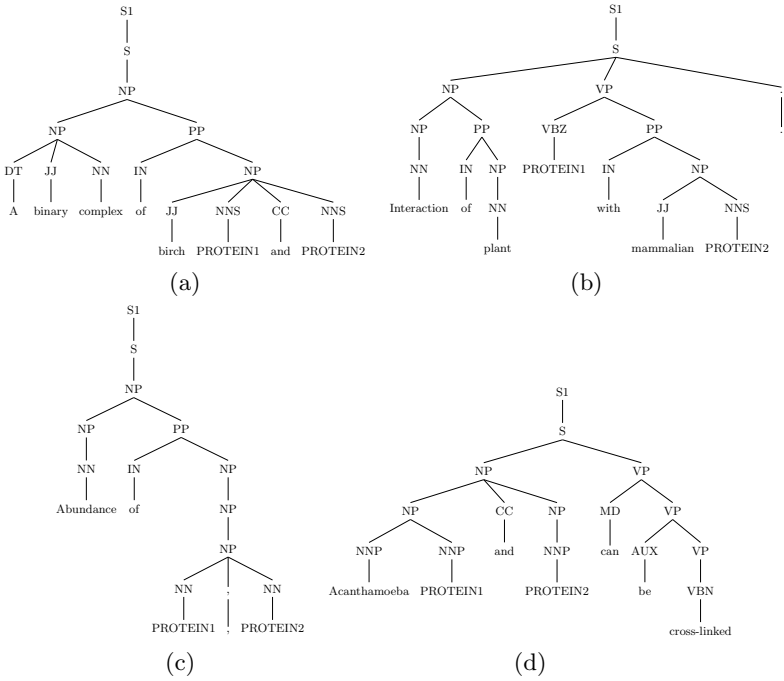

Fig. 2. This figure shows four parsed sentences

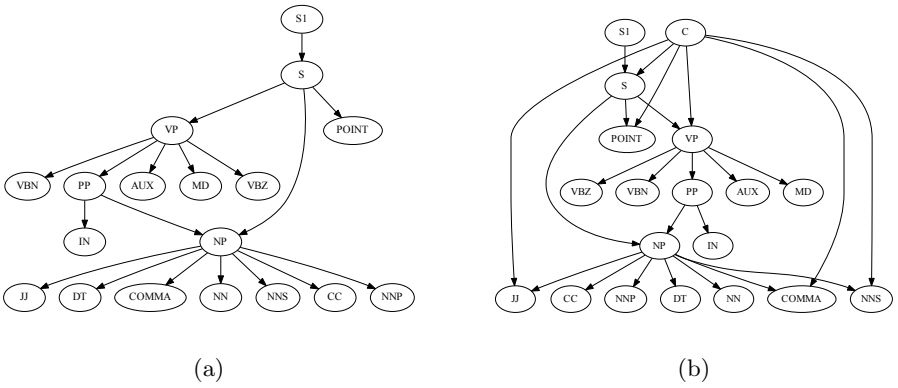

Fig. 3. (a) An acyclic graph containing non-terminal and pre-terminal nodes from the parse trees in Figure 2. (b) A Bayesian network structure with non-terminal and pre-terminal nodes and a class node $\mathcal{C}$.

For each node z_k, we rank the edges $e_{kj} = \left(z_k, z_{\pi_{kj}}\right)$, where $e_{kj} \in E_{\mathcal{M}}$, in a descendent order in terms of $\mathcal{I}$, such that, the first d edges in the rank are considered in the graph and remaining edges are removed.

Finally, in the last step, we add a class node $\mathcal{C}$ and use a hill-climbing local search algorithm and a Bayesian score [15] to connect this class node with the rest of nodes in the graph. This new graph structure is the final Bayesian network structure. For example, Figure 3(b) shows the graph of the Figure 3(a) but including the class node $\mathcal{C}$. The class node $\mathcal{C}$ is a binary variable, where $\mathcal{C} = 1$ means 'true interaction' and $\mathcal{C} = 0$, 'no interaction'.

Parameters of the Bayesian Network. To calculate the parameters θ for each variable z_k $(1 \leq k \leq n)$ in a Bayesian network model, we use the *maximum likelihood estimation* (MLE) [5]. Given a training data set D with M samples, we use the MLE in equation (4).

$$L\left(\hat{\theta} : D\right) = max_{\theta \in \Theta} L\left(\theta : D\right) \tag{4}$$

$$L\left(\theta : D\right) = \prod_{m=1}^{M} P\left(z_1[m], \ldots, z_n[m] : \theta\right) \tag{5}$$

The node $z_k[m]$ corresponds to the z_k feature of the m-th sample. The likelihood estimation of the joint probability $L\left(\theta : D\right)$ given the parameters θ is specified in equation (5). We can decompose this joint probability into shorter and separate terms according to their conditional probabilities in equation (6) [5].

$$L\left(\theta : D\right) = \prod_{m=1}^{M} P\left(z_1[m] \mid z_{\pi_1}[m] : \theta\right) \ldots P\left(z_n[m] \mid z_{\pi_n}[m] : \theta\right) \tag{6}$$

Once we have the structure of the BN and its parameters, the BN model can be used to extract PPIs. In this case, for a test sentence x, we use the queries $P\left(\mathcal{C}[x] = 1 \mid z_1[x], \ldots, z_n[x]\right)$ and $P\left(\mathcal{C}[x] = 0 \mid z_1[x], \ldots, z_n[x]\right)$, in order to calculate the probabilities for the values $C[x] = 1$ and $C[x] = 0$. To infer if there exist an interaction or not, we calculate the maximum a posteriori in equation (1), of the probabilities $P\left(\mathcal{C}[x] = 1 \mid z_1[x], \ldots, z_n[x]\right)$ and $P\left(\mathcal{C}[x] = 0 \mid z_1[x], \ldots, z_n[x]\right)$. Finally, we extract a PPI interaction according to the equation (2).

3.2 Pruning Parse Trees

We employed the *path-enclosed tree method* to prune parse trees [16]. The generalization provided by the pruning has been demonstrated to improve the performance of extraction methods [3]. This method only considers the information surrounding the pair of protein names that are being analysed.

Figure 4 shows an example of pruning using the path-enclosed tree method. Note that after pruning, the leaves nodes "PROTEIN1" and "PROTEIN2" are the borders of the new parse tree.

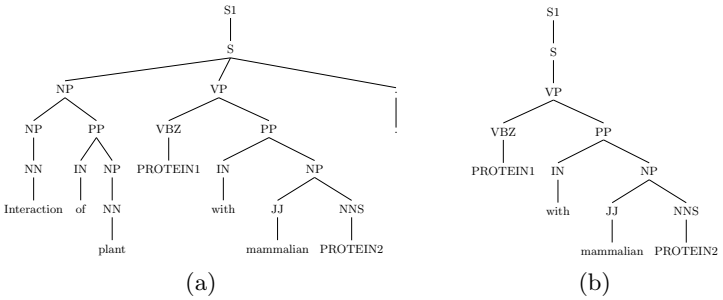

Fig. 4. (a) A parse tree from a sentence ID="BioInfer.d1.s2". (b) A pruned parse tree using the path-enclosed tree method.

4 Experimental Results

4.1 Evaluation Corpora

The proposed method was evaluated on five well-known PPI corpora. The corpora is formed by the following collections: AIMed, BioInfer, IEPA, HPRD50 and LLL; which are described elsewhere [9]. Table 1 summarize general information about the corpora, which are organized in an unified XML format [9]. BioInfer and AIMed corpora are greater than the rest of corpora in terms of instances and sentences. They have 9666 and 5834 instances respectively. In contrast, IEPA, HPRD50 and LLL have 817, 433 and 330 instances respectively. Also, sentences in PPI corpora are normalized, since only a pair of proteins can have an interaction and this interaction can be either positive or negative.

Table 1. General information of the Five PPI Corpora. "# inst. pos" and "# inst. neg" are the number of positive and negative sentences respectively.

	AIMed	BioInfer	HPRD50	IEPA	LLL
# documents	225	836	43	200	45
# sentences	1955	1100	145	486	77
# instances	5834	9666	433	817	330
# inst. pos	1000	2534	163	335	164
# inst. neg	4834	7132	270	482	166

The pair of protein names, forming an instance, is replaced by the following pair of alias: "PROTEIN1" and "PROTEIN2". The rest of protein names are replaced by "PROTEIN". This replacement can be observed in sentences of the Figure 2.

4.2 Experimental Settings

The Charniak parser[2] was used to obtain the parse trees from sentences. In the modelling of Bayesian networks, we extended the Weka 3.6 package [17] in order to employ the parse trees as training data. Furthermore, we use the inference algorithm of Weka over our BN models. In the evaluation of the models, we used the macro-averaged F-measure (ma-F) and employed a 10-fold cross-validation at document level [3].

4.3 Performance of the PPI Extraction

The results are shown in Table 2. BN models have a clear pattern, their performances vary according to the size of the training corpus and the test corpus used. Thus, our models overcome the previous approaches for the larger collections AIMed and BioInfer in comparison with the following best results of CM (2010) [3]. However, their performance is reduced when dealing with smaller corpora, such as HPRD50 and LLL.

Table 2. Results on PPI Corpora. PT-BN: our method. It uses the d parameter to limit the maximum number of parents for each node in the BN. CM (2010): [3]. Miwa (2009a): [10]. ma-F: Average of F-measure. σ_{ma-F}: standard deviation of F-measure.

	PT-BN $d = \infty$		PT-BN $d = 3$		CM (2010)		Miwa (2009a)	
	ma-F	σ_{ma-F}	ma-F	σ_{ma-F}	ma-F	σ_{ma-F}	ma-F	σ_{ma-F}
AImed	68.6	6.0	**76.4**	3.4	67.0	4.5	60.8	6.6
BioInfer	71.9	5.2	**72.8**	4.7	72.6	2.7	68.1	3.2
HPRD50	50.9	12.8	53.1	11.1	**73.1**	10.2	70.9	10.3
IEPA	68.2	4.6	67.4	0.5	**73.1**	6.0	71.7	7.8
LLL	55.43	1.7	59.0	19.3	**82.1**	10.4	80.1	14.1

We tested two types of models: *BN-PT ($d = 3$)* and *BN-PT ($d = \infty$)*. We selected an arbitrary number of parents, $d = 3$, since it proportionally increases the complexity of the search best BN structure. The complexity of the search procedure of the best BN is critical when $d \geq 2$ [5]. In addition, we consider all the parents of a node after our method generates the acyclic directed graph (i.e. without using the mutual information measure to remove edges). In this case, we denote $d = \infty$ to indicate that no edges were removed and all the parents were considered.

It turns out that the number of parents d influences the performance of the BN model. It is logical since the number of parents determines consequently the number of parameters in the BN. To calculate these parameters, the number of instances, in the training data set, must be large enough to learn the BN model. Thus, the results of our BN models in large corpora, such as BioInfer

[2] `http://www.cs.brown.edu/people/ec/#software`

Table 3. Results in cross-corpora. The comparison is performed in terms of F-measure. PT-BN: our method. d: maximum number of parents for each node in the BN. CM: [3]. Miwa (2009a): [10]. Airola (2008): [2]. Training corpora are distributed in rows and test corpora in columns.

	Methods	BioInfer	AIMed	IEPA	HPRD50	LLL
BioInfer	PT-BN $d = 3$	**72.8**	**69.3**	67.4	66.0	65.5
	CM (2010)	72.6	65.2	**72.9**	**71.9**	**78.4**
	Miwa (2009a)	68.1	49.6	71.4	68.3	76.9
	Airola (2008)	61.3	47.2	68.0	63.9	78.0
AIMed	PT-BN $d = 3$	**64.3**	**76.4**	47.4	54.4	39.5
	CM (2010)	**64.2**	67.0	59.0	**72.9**	62.7
	Miwa (2009a)	53.1	60.8	**68.1**	68.3	73.5
	Airola (2008)	47.1	56.4	67.4	69.0	**74.5**
IEPA	PT-BN $d = 3$	**67.3**	**64.2**	67.4	61.9	63.0
	CM (2010)	66.1	57.8	73.1	66.3	78.4
	Miwa (2009a)	55.8	40.4	71.7	66.5	**83.2**
	Airola (2008)	51.7	39.1	**75.1**	**67.5**	77.6
HPRD50	PT-BN $d = 3$	65.0	**67.0**	54.3	53.1	56.8
	CM (2010)	**65.5**	63.1	**69.3**	**73.1**	**73.7**
	Miwa (2009a)	48.6	43.9	67.8	70.9	72.2
	Airola (2008)	42.5	42.2	65.1	63.4	67.9
LLL	PT-BN $d = 3$	58.7	43.6	51.5	54.8	59.0
	CM (2010)	**64.4**	**55.9**	**71.4**	**69.4**	82.1
	Miwa (2009a)	48.90	38.60	65.6	64.0	**83.2**
	Airola (2008)	42.50	33.30	64.9	59.8	76.8

or AIMed, are competitive (and even better) compared to the state-of-the-art methods. Nevertheless, the performance of our BN models decrease with small corpora.

Results using cross-corpora experiments are shown in Table 3. It is worthy of note that the size of the corpus affects greatly the performance of the BN in terms of F-measure. Our method achieves a competitive performance, or even better, when it is used with IEPA, BioInfer or AIMed corpus as training data set and, BioInfer or AIMed corpus as test data set. Note that IEPA, BioInfer and AIMed have more than 480 sentences (486, 1,100 and 1,955 sentences respectively). On the other hand, the corpora HPRD50 and LLL have less than 150 sentences (145 and 77 sentences respectively). The diversity of sentences in large corpora could explain why our models performs better with them.

5 Conclusion

We conclude that the use of non-terminal nodes and their relationships from parse trees provide important information in order to learn BN models for the PPI extraction task.

Thus, the BN-PT method was proposed to learn Bayesian network models from parse trees. Like [3], we pruned the parse trees. Next, we use the parse trees as training data set to learn BN models. These BN models obtained good results in the PPI extraction task when they were used in large corpora such as AIMed and BioInfer. However, the performance considerably decreased in comparison with the state-of-the-art methods when they were employed in small corpora. To overcome this problem, one strategy would be tuning the number of parents d, obtaining a different value according to the corpus used. Nevertheless, taking into account that collections of biomedical texts tend to become larger, the proposed method can be useful in the task of extracting PPIs from these collections.

Acknowledgments. We acknowledge the Brazilian funding agencies CNPq and FAPESP for supporting this research.

References

1. Chowdhary, R., Zhang, J., Liu, J.S.: Bayesian inference of protein-protein interactions from biological literature. Bioinformatics 25(12), 1536–1542 (2009)
2. Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., Salakoski, T.: A graph kernel for protein-protein interaction extraction. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. BioNLP 2008, Stroudsburg, PA, USA, pp. 1–9. Association for Computational Linguistics (2008)
3. Choi, S.P., Myaeng, S.H.: Simplicity is better: revisiting single kernel ppi extraction. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, Stroudsburg, PA, USA, pp. 206–214. Association for Computational Linguistics (2010)
4. Rosario, B., Hearst, M.A.: Multi-way relation classification: Application to protein-protein interactions. In: Proceedings of HLTNAAC05 (2002)
5. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
6. Bui, Q.C.C., Katrenko, S., Sloot, P.M.: A hybrid approach to extract protein-protein interactions. Bioinformatics 27(2), 259–265 (2011)
7. Miwa, M., Saetre, R., Miyao, Y., Tsujii, J.: Entity-focused sentence simplification for relation extraction. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, Stroudsburg, PA, USA, pp. 788–796. Association for Computational Linguistics (2010)
8. Leitner, F., Mardis, S.A., Krallinger, M., Cesareni, G., Hirschman, L.A., Valencia, A.: An overview of biocreative ii.5. IEEE/ACM Trans. Comput. Biol. Bioinformatics 7(3), 385–399 (2010)
9. Pyysalo, S., Airola, A., Heimonen, J., Bjorne, J., Ginter, F., Salakoski, T.: Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics 9, S6 (2008)
10. Miwa, M., Saetre, R., Miyao, Y., Tsujii, J.: A rich feature vector for protein-protein interaction extraction from multiple corpora. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, Stroudsburg, PA, USA, vol. 1, pp. 121–130. Association for Computational Linguistics (2009)

11. Miwa, M., Saetre, R., Miyao, Y., Tsujii, J.: Protein-protein interaction extraction by leveraging multiple kernels and parsers. International Journal of Medical Informatics 78(12), e39–e46 (2009)
12. Liu, B., Qian, L., Wang, H., Zhou, G.: Dependency-driven feature-based learning for extracting protein-protein interactions from biomedical text. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. COLING 2010, Stroudsburg, PA, USA, pp. 757–765. Association for Computational Linguistics (2010)
13. Kim, S., Yoon, J., Yang, J.: Kernel approaches for genic interaction extraction. Bioinformatics 24(1), 118–126 (2008)
14. Rinaldi, F., Schneider, G., Kaljurand, K., Clematide, S., Vachon, T., Romacker, M.: Ontogene in biocreative ii.5. IEEE/ACM Trans. Comput. Biol. Bioinformatics 7(3), 472–480 (2010)
15. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Mach. Learn. 9, 309–347 (1992)
16. Zhang, M., Zhou, G., Aw, A.: Exploring syntactic structured features over parse trees for relation extraction using kernel methods. Inf. Process. Manage. 44(2), 687–701 (2008)
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco (2005)