A Hierarchy and Probability-Based Approach for Inferring AS Relationships

Binbin Liao
State Key Lab. of Software Development Environment
Beihang University
Beijing 100191, P.R.CHINA
liaobinbin@nlsde.buaa.edu.cn

Liandong Liu
State Key Lab. of Software Development Environment
Beihang University
Beijing 100191, P.R.CHINA
lld@nlsde.buaa.edu.cn

Liang Wang
State Key Lab. of Software Development Environment
Beihang University
Beijing 100191, P.R.CHINA
wangliang@nlsde.buaa.edu.cn

Hui Zhang
State Key Lab. of Software Development Environment
Beihang University
Beijing 100191, P.R.CHINA
hzhang@nlsde.buaa.edu.cn

Ke Xu*
State Key Lab. of Software Development Environment
Beihang University
Beijing 100191, P.R.CHINA
kexu@nlsde.buaa.edu.cn

ABSTRACT

The commercial relationships between Autonomous Systems (ASes) are of great importance to understand the Internet reachability and calculate the AS-level paths. Several algorithms have been proposed to solve the AS relationship inference problem and applied to the data of IPv4 network. In assuming that the provider is typically larger than its customers, and the peers usually have comparable sizes, the suggested algorithms exploit the AS degree information to infer AS relationships. In analysis of the AS relationships in the IPv6 network, however, we find that quite a few of the inference results induced by the present approaches are different from the inferences in the IPv4 network. With respect to this observation, we analyze the root cause of the discrepancy and propose an algorithm which combines the AS hierarchy information, an inherent nature of the Internet structure that we can hardly neglect while analyzing the AS relationships, with the optimization model of Type-of-Relationship (ToR) problem to infer the AS relationships more realistically and stably. In this paper, we first present a methodology to classify ASes into four hierarchies, and then use the AS hierarchy information to infer AS relationships. By taking advantage of these partial AS relationship information, we introduce an improved algorithm to solve the ToR problem for the remaining AS pairs. The experimental results support our algorithm in two aspects. On one hand, the comparison with previous works in the IPv4 network shows that most of our inferring AS relationships are consistent with their inferences, while more inferences of our approach are confirmed by the export policies stored in the Internet Routing Registry (IRR) databases. On the other hand, 94.82% of our inference relationships in the IPv6 network are consistent with the inferences in the IPv4 network, which illustrates that our algorithm is more stable than previous algorithms.

*Corresponding author. Email: kexu@nlsde.buaa.edu.cn.

AINTEC'09, November 18–20, 2009, Bangkok, Thailand.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing protocols; C.2.5 [**Local and Wide-Area Networks**]: Internet

General Terms

Algorithm, Experimentation, Measurement

Keywords

Commercial Relationships, Hierarchies, Probability, IPv6

1. INTRODUCTION

The Internet is composed of thousands of autonomous systems (ASes). An AS is a connected collection of IP routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet. The ASes exchange reachability information by using Border Gateway Protocol (BGP) [1]. BGP allows each AS to choose its own policy on selecting the best routes, announcing and accepting routes. The commercial relationships between ASes is one of the most important factors in determining the routing policies and the Internet reachability. It is important and essential for the researchers to obtain the accurate commercial relationships among ASes for further analyzing the Internet.

The following is a description of the most common commercial relationships and the export routing policies that are usually used to implement them [2]:

(1)*Provider-to-Customer (p2c)*: AS_c is a customer of AS_p if AS_c pays AS_p for obtaining connectivity to the rest of the Internet. The export policies are usually as follows: AS_c can export its internal routes and customer routes to its provider AS_p, but usually does not export its other provider or peer routes. AS_p can export its internal routes and customers routes as well as its provider and peer routes to AS_c.

(2)*Sibling-to-Sibling (s2s)*: Two ASes are siblings if they have mutual transit agreement, usually for backup. The export policies are usually as follows: each of the two AS can export its internal routes, its customers as well its provider or other routes to the other.

(3)*Peer-to-Peer (p2p)*: Two ASes are peers if they mutually agree to exchange traffic between their customers, often free of charge. The export policies are usually as follows: each of the two AS can export its internal routes and customer routes, but usually does not export its provider or peer routes.

Although several algorithms [3–9] have been proposed in the literature for inferring AS relationships, we find that the performances of them are more susceptible to the network topology variation, which results that the stability of those algorithms is far from satisfactory. Specifically, when we compare the inferring AS relationships of the IPv4 network with the one of the IPv6 network, we observe that quite a few of the inference results are inconsistent. Some ASes owned by the well-known global providers are inferred as customers of the ASes operated by companies that do not provide connectivity services in the IPv6 network. To solve this problem, we propose a new AS relationship inference approach which aims to improve current methods by combining the AS hierarchy information with the optimization model of the Type-of-Relationship (ToR) [5] problem.

The main idea behind this paper is that the Internet is a hierarchical structure network. The core backbone network is composed of the Internet service providers (ISPs) which are companies that offer Internet connection services to their customers. The set of ASes near the core of the network are more likely to be the providers than the other ASes near the periphery of the network that purchase IP transit from ISPs to reach some portion of the Internet. Therefore, the AS hierarchy information plays a crucial role in shaping the AS relationships. The hierarchy of ASes is subject to the network characteristics, external connectivity patterns, the ranges of network services, etc. However it mainly depends on the types of the organizations that operate the ASes. For example, the telecommunications companies are usually the Internet service providers while the university or college networks are typically customers. Since the AS hierarchy is closely related to the types of organizations, differing from AS degrees, it is not affected by the topology graph change. Hence, instead of solely relying on the AS degree information, we also take advantage of the AS hierarchy information in our inference approach to improve the robustness.

In this paper, we first classify ASes into four hierarchies: large ISPs, small ISPs, Internet exchange points (IXPs) and customer ASes, and use the AS hierarchy information to infer most of the AS relationships and determine the structures of the AS paths. For the remaining AS pairs, we introduce an improved algorithm based on combinatorial optimization to compute the types of relationships of them. Furthermore, we find that the way of assigning weights in MAX2SAT formulation introduced in [8] is inconsistent, which might cause the problem of the same clause having two different weights. To solve this problem, we propose a uniform way based on probability to assign weights. Our algorithm is tested on the data obtained from Route Views router [10] in both of the IPv4 and IPv6 networks, and it successfully infers edges in the AS topology graphs with smaller proportion of invalid paths. The experimental results show that our algorithm not only provides an accurate and stable inference result but also captures the realistic Internet hierarchy.

The rest of this paper is structured as follows. In Section 2, we introduce an overview of previous work on AS relationship inference. Section 3 describes our AS hierarchy classification methodology. In Section 4, we present our algorithm for inferring AS relationships. Section 5 shows the results of our inference and the comparison between our algorithm and the existing algorithms. We conclude the paper in Section 6.

2. RELATED WORK

Gao [3] was the first to study the AS relationships problem. Assuming that a provider had a larger size than its customers and every AS path must comply with the valley-free model, Gao proposed a heuristic inference algorithm that identified top-providers and peering links based on AS degrees. In the follow-up studies, Xia and Gao [4] presented a new inference algorithm by using the partial information regarding AS relationships. However, this partial information obtained from Internet Routing Registry IRR [11] databases is usually considered out-of-date.

In [5], Subramanian *et al.* formulated AS relationship inference as an optimization problem, *Type-of-Relationship* (ToR) problem. The authors obtained routing data from various looking glasses called vantage points in [5], and for each vantage point, they assigned a rank to the ASes and then assigned relationships by comparing the basis of the ranks. If two adjacent ASs had different ranks, the one with lower rank was considered as a customer and the other as a provider. If the ranks were similar, the ASs were considered as peers.

Di Battista *et al.* [6] and Erlebach *et al.* [7] independently proposed a similar approach to infer AS relationships. Their algorithms were based on a reduction of ToR problem to the well-known Boolean Satisfiability Problem (SAT). The authors respectively found strict solutions to ToR problem and made a straightforward observation that peering edges could not be inferred in the ToR problem formulation. Although the optimization model insured the number of paths that violate the valley-free path model was minimized, the inference results were far away from the reality [8].

Cohen and Raz [9] pointed out that the current solutions to ToR problem failed to capture the hierarchical structure of the AS graph, which brought the result that the Internet directed graph imposed by AS relationships contained cycles. To solve this problem, they redefined the formal ToR problem to the *Acyclic Type-of-Relationship*(AToR) problem and proposed an algorithm to solve it while keeping the directed graph acyclic. The definition of the AToR problem partially overcame the limitation in the optimization model by taking hierarchical structure into consideration. However, the orientations of edges without causing cycles were not exclusive and might lead to unrealistic AS hierarchies.

Dimitropoulos *et al.* [8] extended the combinatorial optimization approach by incorporating AS-degree-based information into the problem formulation. However, they did not take AS hierarchy information into account and only used degree gradient to measure the relationships between ASes. Furthermore, in their problem formulation, the way of assigning weights to 1-link clauses and 2-link clauses were inconsistent, which might cause the problem of the same clause having two different weights. For example, the two clauses $x_1 \vee \overline{x_2}$ (with $x_2 = 1$) and x_1 are essentially the same but they may have two different weights in their problem formulation. To solve the problem of inconsistency in weight assignment, we present a new method for assigning weights based on probability.

The existing AS relationship inference algorithms usually take a list of AS paths obtained from one or more BGP tables as input and produce a relationship assignment as output. A complete AS topology is the precondition of their inference, because they generally hold the following assumptions: the AS topology graph obtained from BGP tables is close to the reality; a provider typically has a larger size than its customers and the peers usually have comparable sizes; the AS size is usually proportional to its degree. However, since the size of IPv6 network is much smaller than the size of IPv4 network, a lot of the small Internet service providers have small degrees, which results that the AS degrees in the topological graph can not fully reflect the AS sizes. In fact, quite a few of the small ISPs have the degree of 1 or 2, which is also typical for the customer ASes. The dependence on the network topology results that the existing solutions fail to provide a stable inference result in both of the IPv4 and IPv6 networks. These observation highly motivate our work, driving us to seek an algorithm which can produce an AS relationship inference result closer to the reality and is also robust to the network topology change.

3. THE METHODOLOGY FOR CLASSIFYING AS

In the Internet, the ASes are operated by many different administrative domains such as Internet service providers (ISPs), small private companies and universities. The types of organizations mainly determine the ASes hierarchies. For example, the telecommunications companies are at a higher hierarchy level since they are typically the ISPs, while the university or college networks are at a lower level. Thus, appropriately classifying the types of the organizations that own the ASes plays a crucial role in shaping the AS relationships. In this section, we describe our methodology for classifying ASes in detail.

3.1 The AS Classes

Previous works on AS classification suggest that the most common types of ASes include Internet service providers (ISPs), Internet exchange points (IXPs), Network Information Centers (NICs), universities or colleges, research and education centers, and private companies. In the view of hierarchy, we divide the ASes into four different classes as follows:

C_1: Large ISPs (LISPs) - they are the collection at the top of the AS hierarchy. The rest of the ASes rely on one or more large ISPs to reach part of the Internet directly or indirectly.

According to the characteristics of the large ISPs, we can conclude that they are usually the large backbone providers, well-known telecommunications companies, or tier-1 ISPs with intercontinental networks. Hence, we construct this class in a way which differs from the following. We chose the largest telecommunications companies in various countries such as Level 3 Communications, Nippon Telegraph and Telephone Corporation, Tata Communications Limited, etc. to be large ISPs. Since the large telecommunications companies are a relatively small and stable group, accurately finding out the top-level ASes is fairly easy.

C_2: Small ISPs (SISPs) - they are the regional or access providers which offer customers access to the Internet and related services. C_2 includes all the ISPs not in C_1.

C_3: Internet exchange points (IXPs) - they are physical infrastructures that allow different ISPs (the ASes in the first two classes) to exchange traffic between their networks by means of mutual peering agreements, which allow traffic to be exchanged without cost [12]. In particular, a large percentage of ISPs in C_2 have a significant number of IXP connections [13].

C_4: Customer ASes (Custs) - they are usually universities, small private companies or government administrations that have their own networks but do not provide Internet connection services. They pay their providers for connecting to the rest of the Internet.

3.2 The Results of Classification

In the Internet Routing Registry (IRR) [11] databases, the descriptions or the names of the organizations responsible for the AS numbers are stored in the *descr* attribute of the RPSL [14] *aut-num* class. The following is the example of descr attributes of AS1 and AS2 respectively, "LVLT-1 - Level 3 Communications, Inc." and "DCN-AS - University of Delaware". Although the descr attribute does not have a standard representation, each class has some common words or phrases which represent the characteristics of organizations in the class. For example, the words "isp","ix" and "univ." describe the Internet service providers, Internet exchange points and universities respectively. However, finding those keywords manually is inefficient and incomprehensive. To classify the organization records effectively, we build our classification method by using Support Vector Machine [15] which has a good performance in text multi-classification. To reduce the number of features, we preprocess the organization descriptions by removing stop words such as "the","an","of" and the words with little meaning such as numbers and country names.

Our data is collected on 03/14/2009 from the CIDR Report [16] which providers the integrated organization description records extracted from RIPE NCC, APNIC, ARIN, etc. IRR databases. After removing records of private ASes and unregistered ASes, we obtain 46,186 organization records of ASes in the IPv4 network and 1420 records in the IPv6 network. Since the same AS numbers in the IPv4 network and the IPv6 network represent the same organizations, we combine the organization records in both IPv4 and IPv6 networks and finally obtain 46,186 records, i.e. the ASes in the IPv6 network is virtually a subset of the ASes in the IPv4 network. We construct the C_1 class by adding the records corresponding to the largest telecommunications companies in various countries, and then randomly select 1000 records in the remaining to be our training set. In addition, we add 15 well-known IXPs into our training

Table 1: Number and Proportion in Each Class.

		C_1	C_2	C_3	C_4
AS	IPv4	129	15,537	267	30,217
	IPv6	36	1568	26	790
%	IPv4	0.28%	33.72%	0.58%	65.42%
	IPv6	2.54%	40.00%	1.83%	55.63%

Table 2: Distributions of Training Set,Validation Set and Predicted Result.

	C_2	C_3	C_4
Training set	303	16	696
Validation set	299	0	716
Veracious predicted result	251	0	582

set to ensure the IXPs records included. For each record, we manually determine its correct class by examining its organization characteristic and searching for references [17–20] to it. Having the training set prepared, we use LibSVM [21] a publicly available implementation tool for SVM to realize our classification. We omit the details of setting the parameters of SVM for the sake of brevity.

After applying the classifier obtained from LibSVM to the remaining records, we get the classification results shown in Table 1. Among the classified ASes in the IPv4 (IPv6) network, 0.28% (2.54%) are large ISPs, 33.72% (40.0%) are small ISPs, 0.58% (1.83%) are IXPs, and 65.42% (55.63%) are customers. Since IPv6 is in its infancy in terms of general worldwide development, it is mainly deployed by the ISPs. So the percentage of ISPs of IPv6 network is relatively higher. Furthermore, we find that more small ISPs in the IPv6 network than in the IPv4 network have the AS degree of 1 or 2, which indicates that, unlike the IPv4 network, the AS degree in the IPv6 network is not utterly proportional to the AS size. To validate our result, 1000 records in the result set are chosen at random and their correct classes are manually determined. For each record, we compare the correct class and the class predicted by our classifier. The result shows that 833 records are classified correctly. To a certain extent, it demonstrates that our classification is effective. The distributions of the training set, the validation set and the predicted result are shown in Table 2.

It is well-known that the Internet AS topology has an inherent hierarchical structure, arising from the difference between ASes. The provider AS typically should be at a higher level than its customer AS in the hierarchy. Based on this idea, we incorporate the AS classification results into our AS relationship inference algorithm which is described in detail in the next section.

4. THE ALGORITHM FOR INFERRING AS RELATIONSHIPS

Since the AS hierarchy plays a vita role in our inference algorithm, prior to presenting the relationship inference algorithm, we explain how the AS hierarchy information is used to infer the AS relationships and determine the structure of the AS paths. We denote an edge from an AS at a lower hierarchy level to an AS at a higher hierarchy level with a -1, the edge in a reverse direction with a 1, and an edge between the ASes in the same hierarchy level with a 0.

(1) In filtering the abnormal AS paths, we use the AS hierarchy information to detect whether a path violates the valley-free model. In a path, if an AS at a lower level appears between the ASes at a higher level, then the path should be considered as having violated the hierarchical structure. Namely, there is at least one edge labeled with 1 appears before an edge labeled with -1 in a path. We carefully examine the AS pairs which lead to the abnormities to detect whether they are caused by false classification. Among those AS pairs, there are 6.70% of them raised by misclassification, which indicates that the AS hierarchy introduces minimal inference mistakes. We filter the remaining abnormal paths, because they are most likely to be caused by BGP misconfigurations or special routing policies, filtering those paths helps to mitigate the impact on inference results.

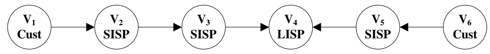

Figure 1: A path contains six vertices with their corresponding hierarchies (the direction is from customer to provider).

(2) In inferring the p2c or c2p relationships, we identify the relationships of edges by comparing the hierarchies of the adjacent ASs. For example, consider the path contains 6 ASes with their hierarchies depicted in Fig. 1. The corresponding edges have a label set {-1, 0, -1, 1, 1} from left to right. We find out the edges which correspond to the last -1 and the first 1 appearing in the path, and refer these edges as *lastC2P* and *fisrstP2C* respectively. Assuming the path is valid, we can conclude that the edges before *lastC2P* should be labeled with -1 and the edges after *fisrstP2C* should be labeled with 1. Specifically, in Fig. 1, since the edge from ASv_2 to ASv_3 appears before the edge labeled with -1 (from ASv_3 to ASv_4), the relationship of it is inferred to be in the c2p type. If the edges between the adjacent ASes cannot be inferred in this stage, we will initially orient them along the node degree gradient and utilize the algorithm described in Section 4.2 to infer their relationships.

(3) In inferring the p2p relationships, since the adjacent ASes in the same hierarchy level and the ASes participating at the IXPs are most likely to have the p2p relationships, we treat them as peer candidates and use the algorithm described in Section 4.3 to infer the p2p relationships within the candidate set.

4.1 Inferring the S2S Relationships

Since s2s relationships exist between the adjacent ASes belonging to the same organization, a natural approach to infer s2s relationships is to divide ASes into different groups according to their registered organizations, and then examine whether the adjacent ASes of each AS path are in a same group. Specifically, we utilize the AS organization information stored in IRR [11] databases to create groups of sibling ASes similar to the way described in [8]. The strength of this method for identifying s2s relationships relies on that it takes advantage of the AS organization information which is believed to be accurate and reliable.

4.2 Inferring the P2C Relationships

The problem of inferring p2c relationships among ASes can be formulated as the Type-of-Relationship (ToR) problem [5]: given a set of AS paths P and an undirected graph G obtained from P, orient all edges in G so that a maxi-

Table 3: Mapping between AS Edges and 2SAT Clauses [6].

A directed path	2SAT clause
$V_1 \xrightarrow{x_1} V_2 \xleftarrow{x_2} V_3$	$x_1 \vee x_2$
$V_1 \xrightarrow{x_1} V_2 \xrightarrow{x_2} V_3$	$x_1 \vee \overline{x_2}$
$V_1 \xleftarrow{x_1} V_2 \xleftarrow{x_2} V_3$	$\overline{x_1} \vee x_2$
$V_1 \xleftarrow{x_1} V_2 \xrightarrow{x_2} V_3$	$\overline{x_1} \vee \overline{x_2}$

mum number of paths are valid. Battista *et al.* [6] proposed an approximation algorithm to solve the ToR problem by reducing it to a SAT formula (The mapping between AS edges and 2SAT clauses is shown in Table 3). Dimitropoulos *et al.* [8] showed that the optimal solution of ToR problem might lead to unrealistic AS relationship inferences, and proposed a method to reformulate the ToR problem as a multi-objective optimization problem (MAX2SAT) by introducing AS degree information.

However, there exists a deficiency with respect to the weight assignment in the problem formulation of the method in [8]. Specifically, the 1-link clauses are weighted by a function of the node degree gradient $f(d^-, d^+)$, while the weights of 2-link clauses are equal. The difference between 2-link clauses is not reflected by their weights. The way of assigning weights to 1-link clauses and 2-link are inconsistent, which may cause the problem of the same clause having two different weights. For example, the two clauses $x_1 \vee \overline{x_2}$ (with $x_2 = 1$) and x_1 are essentially the same but they may have two different weights in the problem formulation. Furthermore, since the value of α for controlling the tradeoff between 1-link clauses and 2-link clauses has an important impact on the inference results, we must fix its value appropriately. However, it is difficult to find the value of α that fits the dynamically changing topology data, unless we know the realistic AS relationships in advance. The necessity of fixing the value of α inevitably limits the universality of the algorithm. Therefore we propose a new way which is based on probability to assign weights to 1-link and 2-link clauses, so that when one of the two variables in a 2-link clause has been identified, i.e. the relationship of the corresponding edge has been classified by comparing the hierarchies of the adjacent ASs, the 2-link clause would be converted into a 1-link clause while their weights keep consistent according to their probability (detailed in Theorem 1).

Now we use a path which includes 5 ASes as an example to explain our weight assignment method (For brevity, we omit the details about processes of solving the MAX2SAT problem by using semidefinite programming relaxation). If the ASes with their types in the path accord with Fig. 2(a), by comparing the hierarchies of the adjacent ASs, we can deterministically classify the relationships between those ASes (the direction is from customer to provider). If there are some ASes in the same hierarchy level, for example *ASv2*, *ASv3* and *ASv4* in Fig. 2(b), and we can only identify a portion of relationships within the ASes, i.e. *ASv1* and *ASv2*, *ASv4* and *ASv5*. Since the known relationships are not cared in the following procedure, we remove the corresponding edges and simplify the path into the one depicted

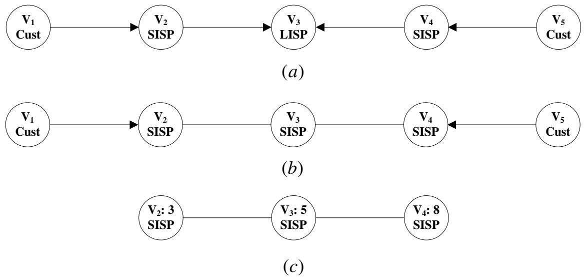

Figure 2: A path contains five ASes with their types and degrees.

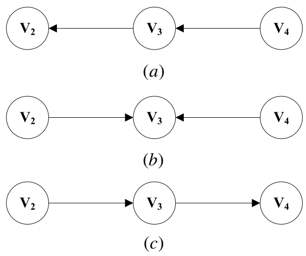

Figure 3: Examples of orientations which make the path in Fig. 2(b) valid.

in Fig. 2(c). According to Gao's valid AS path model [3], if the path depicted in Fig. 2(b) is a valid path, the directions of edges in Fig. 2(c) can only be one of the 3 cases shown in Fig. 3.

Given the following definitions:

$Deg(v_i)$ denotes the degree of ASv_i.

TotalDeg denotes the sum of the degrees of ASes in a path.

$\Pr(v_i) = \frac{Deg(v_i)}{TotalDeg}$ denotes the probability of ASv_i being the top-provider in a path.

$\Pr(v_i, v_j)$ denotes the probability of ASv_i being the customer of ASv_j.

Assuming that the degrees of ASv_2, ASv_3 and ASv_4 are 3, 5 and 8 respectively, the probability of case (a), meaning that the ASv_2 is the top-provider, is:

$$\Pr(v_2) = \frac{Deg(v_2)}{TotalDeg} = \frac{3}{16}$$

Similarly, the probabilities of case (b) and (c) are:

$$\Pr(v_3) = \frac{Deg(v_3)}{TotalDeg} = \frac{5}{16}, \Pr(v_4) = \frac{Deg(v_4)}{TotalDeg} = \frac{8}{16}$$

Since the degree of ASv_4 is largest among the ASes, case (c) in Fig. 3 is more likely to reflect the realistic relationships of the ASes, meaning that ASv_4 shall be the top-provider on this path. Therefore the relationship between ASv_2 and ASv_3 and the relationship between ASv_3 and ASv_4 shall be in c2p types. Case (b) and case (c) include the instance of the edge from ASv_2 to ASv_3, so the probability of the edge from ASv_2 to ASv_3 is:

$$\Pr(v_2, v_3) = \Pr(v_3) + \Pr(v_4) = \frac{13}{16}$$

Similarly, the probability of the edge from ASv_3 to ASv_4 is:

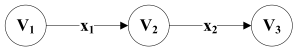

Figure 4: An example of a 2-link clause.

$$\Pr(v_3, v_4) = \Pr(v_4) = \frac{8}{16}$$

We map the AS edges to the 2SAT clauses according to the mapping rules in Table 3. The description of the 2SAT problem of case (c) in Fig. 3 is $(x_1 \vee \overline{x_2})$.

We define that the probability of clause $(x_k \vee x_l)$ is:

$$\begin{aligned}\Pr(x_k \vee x_l) &= 1 - \Pr(\overline{x_k \vee x_l}) = 1 - \Pr(\overline{x_k} \wedge \overline{x_l}) \\ &= 1 - [1 - \Pr(x_k)][1 - \Pr(x_l)]\end{aligned}$$

And the weight of the clause $(x_1 \vee \overline{x_2})$ is:

$$\begin{aligned}\mathcal{W}(x_1 \vee \overline{x_2}) &= \Pr(x_1 \vee \overline{x_2}) = 1 - [1 - \Pr(x_1)][1 - \Pr(\overline{x_2})] \\ &= 1 - [1 - \Pr(v_2, v_3)][1 - \Pr(v_3, v_4)]\end{aligned}$$

Our probability-based method solves the problem of inconsistency of assigning weights to 1-link clauses and 2-link clauses of the problem formulation in [8]. The theorem and its proof are presented as below.

THEOREM 1. *For a 2-link clause $l_1 \vee l_2$ (where l_1 and l_2 are either variables or the negation of variables). If l_2 has been identified to be false, then the weight of the 2-link clause is equal to the weight of the 1-link clause l_1.*

PROOF. Without loss of generality, we will only consider the case shown in Fig. 4. The other cases can be proved similarly.

The 2SAT formulation of the case in Fig. 4 can be described as $(x_1 \vee \overline{x_2})$. According to the probability-based weight assignment method, the weight of the clause is:

$$\mathcal{W}(x_1 \vee \overline{x_2}) = \Pr(x_1 \vee \overline{x_2})$$

If the direction between v_2 and v_3 has been identified as from v_2 to v_3, then $\Pr(x_2)$=1. Putting the value of $\Pr(x_2)$ into the expression above, we have

$$\begin{aligned}\mathcal{W}(x_1 \vee \overline{x_2}) &= \Pr(x_1 \vee \overline{x_2}) = \Pr(x_1 \vee 0) \\ &= \Pr(x_1) = \mathcal{W}(x_1 \vee x_1)\end{aligned}$$

So we finish the proof of Theorem 1. □

In summary, our inference algorithm contains two steps. First, it exploits the AS hierarchy information to determine the structure of the AS paths, which avoids the ISPs being inferred as customers of non-ISPs. In the second step, after removing the deterministic relationships, it reduces the ToR problem to the MAX2SAT problem and uses a probability-based method to assign weights. Unlike the previous work, our algorithm takes both AS degrees and AS hierarchy constrains into consideration and does not contain any tuning parameters, which makes it more competent and universal.

4.3 Inferring the P2P relationships

A solution of the algorithm described in the previous sections determines p2c relationships between the ASes. Since some of the ASes may have p2p relationships, we need to discover those peering ASes, basing on the obtained solution. With respect to the valley-free model, a valid path can have only one peer link adjacent to the top-provider. Hence, a natural way to identify the peer link is to construct the peer candidates, and then examine if the corresponding p2c edge can be converted to p2p edge without causing invalid path and without violating the hierarchical structure of the AS path. Specially, since the adjacent ASes in the same hierarchy level and the ASes participating at the IXPs are most likely to have the p2p relationships, we treat them as our peer candidates, and then convert the edges within the candidate set one by one at the premise of keeping the number of invalid paths minimized.

5. EXPERIMENTAL RESULTS

5.1 Data sources and inference results

In our experiment, we use the real data from Route Views [10] project to examine our inference algorithm. We collect the data from 03/14/2009 to 03/21/2009. After removing AS prepending and AS sets, we obtain 29,354 (1327) ASes, 14,835,569 (14,643) AS paths and 68,912 (4718) links of the IPv4 (IPv6) network.

Before starting the relationship inference, we identify the s2s links and remove them from the AS topology graph. Using the AS hierarchy information, we map the ASes into their types and obtain partial AS relationships by comparing the hierarchies of the adjacent ASes. Basing on the partial relationships, we examine the valley-free property for each AS path and remove the abnormal paths. For the remaining AS links, we map them to a set of clauses by using the reduction of ToR to MAX2SAT, and then assign weights to the clauses in a probability-based way. To obtain better approximation ratio, we use the SDP based approximation algorithm developed by Lewin *et al.* [22] to solve the MAX2SAT problem and the SDP solver SDPT3v4.0 [23] to solve the semidefinite programs. We build the peer candidate set upon the p2c inference results. After examining the peer candidate set, we identify the links in the type of p2p relationship.

Our inference result is shown in Table 4. Within 68,912 (4718) edges in the IPv4 (IPv6) network, there are 319 (74) s2s edges, 62,484 (4066) p2c edges and 7487 (578) p2p edges. Among the p2c edges in the IPv4 (IPv6) network, 61.95% (69.04%) of them are inferred by comparing their hierarchies. The inference result matches the previous work, illustrating that the p2c relationship accounts for most of the AS relationships.

Table 4: Results of Our AS Relationship Inference.

	Edges	S2S	P2C		P2P
IPv4	68,912	319 0.46%	62,484 90.67%	38709 61.95%	7487 10.86%
IPv6	4718	74 1.57%	4066 86.18%	2807 69.04%	578 12.25%

5.2 Validation of Our Method

Since most of the inferring p2c relationships are based on the AS hierarchy, we evaluate the effect that the AS hierarchy has on the inference results by comparing the algorithm incorporated with the AS hierarchy information to the

Table 5: Comparison of the Inference Algorithms (19,497 (1518) common AS pairs exist in both IRR databases and BGP tables. S2S: 1.09%(1.45%); P2C: 55.32%(61.53%); P2P: 43.59%(37.02%)).

		IPv4	Mismatch	IPv6	Mismatch	Inconsistency
Gao	S2S	896	69 (32.55%)	182	11 (50.00%)	36 (19.78%)
	P2C	64,364	498 (4.61%)	4279	72 (7.71%)	404 (9.44%)
	P2P	3652	1339 (12.41%)	257	64 (11.39%)	94 (36.58%)
DPP	P2C	68,912	377 (3.49%)	4718	52 (5.57%)	419 (8.88%)
CAIDA	S2S	211	26 (12.26%)	74	4 (18.18%)	5 (6.76%)
	P2C	63,250	367 (3.40%)	4152	82 (8.78%)	497 (11.97%)
	P2P	7066	778(9.15%)	492	79 (14.06%)	102 (20.73%)
AToR	S2S	489	58 (27.36%)	57	9 (40.91%)	28 (49.12%)
	P2C	61,430	298 (2.76%)	4139	109 (11.67%)	498 (12.03%)
	P2P	6993	849 (9.99%)	522	85 (15.12%)	70 (13.41%)
HPB	S2S	319	23 (7.21%)	74	11 (14.86%)	0 (0.00%)
	P2C	62,484	182 (1.68%)	4066	21 (2.25%)	210 (5.16%)
	P2P	7487	674 (7.93%)	578	32 (5.70%)	34 (5.89%)
PB	S2S	319	23 (10.83%)	74	4 (18.18%)	0 (0.00%)
	P2C	63,009	219 (2.03%)	3912	67 (7.17%)	349 (8.92%)
	P2P	5584	727 (8.55%)	732	59 (10.50%)	76 (10.38%)

original algorithm. Specially, without invoking AS hierarchy information in the original algorithm, we determine the top-provider as the highest degree AS in the path and only use the reduction of ToR to MAX2SAT and give weights to the clauses in a probability-based way detailed in Section 4.2. We refer these two algorithms as HPB (Hierarchy and Probability-Based) and PB (Probability-Based) respectively. The two rows named HPB and PB in Table 5 provide the results of evaluating. It shows that the effect of incorporating AS hierarchy information into the algorithm is less on the IPv4 network than on the IPv6 network. Namely, the number of inconsistent relationships is relatively smaller in the result of HPB algorithm (If the relationship between an AS pair in the IPv4 network is different from the one in the IPv6 network, it means that there is an inconsistency). By analyzing the inconsistency in the other four algorithms, we find that the number of inconsistent relationships is also higher in these algorithms. This observation illustrates that the AS hierarchy information needs to be considered, especially in the IPv6 network.

Validating the inference results is a challenging task hampered by the fact that it is difficult to collect the actual AS relationships which are considered sensitive business information by ISPs. However, we can obtain the partial AS relationships by analyzing the routing policies registered in IRR [11] databases, because AS relationship is one of the most important factors in determining routing policy. In our experiment, we use the AS relationships induced from IRR databases to evaluate our algorithm and the present algorithms. The comparison results are shown in Table 5 (Gao refers to the algorithm presented in [3], DPP refers to the algorithm presented in [5], CAIDA refers to the algorithm presented in [8] and AToR refers to the algorithm presented in [9]). The comparing algorithms are tested on the same AS path set, except CAIDA of the IPv4 network, since it has published the results of the inference on the website [24]. We do not compare with the PTE algorithm in [4] since its inference is based almost completely on the IRR database.

Based on the methods introduced in [4] and [25], we process the data in the IRR databases on 03/14/2009, and obtain the relationships of 45,220 (2761) AS pairs in the IPv4 (IPv6) network. Within these AS relationships, there is a high percentage of p2p links, which matches the observation done in [8], indicating that to capture a veracious of p2p relationships, it is more necessary than exploiting the inference algorithm for us to take advantage of the different data sources, i.e. the IRR databases, because most of the missing links in BGP tables are in the p2p types. The number of the common AS pairs exist in both IRR databases and BGP tables is 19,497 (1518). Among these AS pairs, there are 55.32% (61.53%) of them have the type of p2c relationships, 43.59% (37.02%) are in the p2p relationships and 1.09% (1.45%) are s2s types. Although the AToR algorithm infers the p2p relationships in the way similar to ours, it does not use filtering but examined all the paths to check whether the conversion of p2c links to p2p links cause invalid paths, which results in an exceeding long runtime if the set of input paths is large. Since the DPP algorithm does not concern the s2s and p2p relationships, the corresponding columns are omitted. After comparing the relationships inferred from the two d data sources, we find that the number of mismatching relationships is relatively smaller in HPB (If an AS relationship inferred from the algorithm is different from the one induced from IRR databases, it means that there is a mismatch), which indicates that the AS relationships inferred by HPB has greater consistency with the export policies stored in IRR databases than the other algorithms.

Since the records in IRR databases are not necessarily up-to-date and the intersection of the AS links in both of the IRR databases and the BGP tables is relatively small, we perform the comparison experiment between our algorithm and the other suggested algorithms to get a better sense of the distribution of the inference relationships in both IPv4 and IPv6 networks. Table 6 shows the number of consistent relationships for each type between our algorithm and the reference algorithms. It shows that in the IPv4 network, there is a high agreement on p2c links between HPB and the other five algorithms. Since HBP and CAIDA adopt a similar method to identify the s2s links, the consistence of this type between them is much higher, while the consistence is lower when comparing with Gao and AToR. With respect

Table 6: Consistent results of Our Inference Algorithm and Other Algorithms.

HPB	IPv4			IPv6		
	S2S	**P2C**	**P2P**	**S2S**	**P2C**	**P2P**
	319	62,484	7487	74	4066	578
Gao	55	60,154	2841	33	2981	167
PTE	271	62,008	4885	56	3644	397
DPP	–	59,986	–	–	3267	–
CAIDA	211	61,118	3767	74	3455	223
AToR	69	60,169	3321	41	3255	279

to the p2p links, the number of p2p links inferred by HPB and PTE is relatively similar while some conflicts emerge in HBP and Gao. However, there is distinct agreement among the algorithms in the IPv6 network. Even for the p2c type, quite a few AS pairs are inferred in opposite relationships when comparing HBP with other algorithms, except PTE. We find that most of the small ISPs have the AS degree of 1 or 2, which causes them to be inferred as the customers of the non-ISPs with relatively larger degrees. Since the paths containing those AS pairs usually appear only a few times in the IPv6 network, it is hard to filter them by examining the valley-free property and controlling the invalid paths number. This observation supports our point of view that the AS hierarchy is an inherent nature of the Internet structure that we can hardly neglect while analyzing the AS relationships.

6. CONCLUSION

In this paper, we study the AS relationship inference problem. We observe that the current algorithms are not very robust when they are applied to different networks. After analyzing the root cause of this limitation, we propose an algorithm which combines the AS hierarchy information with the optimization model of the ToR problem to infer the AS relationships. In addition, our probability-based weight assignment method solve the problem of inconsistency occurring in previous work. The experimental results show that our algorithm not only achieves good inferring results in the IPv4 network, but also keeps a more stable performance than the present approaches in the IPv6 network.

7. REFERENCES

[1] J. W. Stewart. *BGP4: Inter-Domain Routing in the Internet.* Addison-Wesley Publishing Company, 1999.

[2] G. Huston. Interconnection, peering and settlement. *Internet Protocol Journal*, Jun. 1999.

[3] L. Gao. On Inferring Autonomous System relationships in the Internet. *IEEE/ACM Transactions on Networking (TON)*, Sept. 2001.

[4] J. Xia and L. Gao. On the evaluation of AS relationship inferences. *IEEE Globecom*, Nov. 2004.

[5] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. *IEEE INFOCOM*, Jun. 2002.

[6] G. Di Battista, M. Patrignani, and M. Pizzonia. Computing the Types of the Relationships between Autonomous Systems. *IEEE/ACM Transactions on Networking (TON)*, Apr. 2003.

[7] T. Erlebach, A. Hall, and T. Schank. Classifying customer-provider relationships in the Internet. *In Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN)*, Nov. 2002.

[8] X. Dimitropoulos, D. Krioukov, B. Huffaker, kc claffy, and G. Riley. Inferring AS Relationships: Dead End or Lively Beginning? *In Proceedings of the 4th Workshop on Efficient and Experimental Algorithms (WEA'05)*, May 2005.

[9] R. Cohen and D.Raz. Acyclic type of relationships between autonomous systems. *IEEE INFOCOM*, 2007.

[10] Route Views Project. `http://www.routeviews.org/`.

[11] Internet Routing Registry (IRR). `http://www.irr.net/docs/list.html/`.

[12] Wikipedia, Internet eXchange Point. `http://en.wikipedia.org/wiki/Internet_Exchange_Point/`.

[13] K. Xu, Z.Duan, Z. Zhang, and J. Chandrashekar. On Properties of Internet Exchange Points and Their Impact on AS Topology and Relationship. *The 3rd International IFIP-TC6 Networking Conference*, vol.3042:pp.284–295, Springer Berlin, Apr. 2004.

[14] C. Alaettinoglu, T. Bates, E. Gerich, D. Karrenberg, D. Meyer, M. Terpstra, and C. Villamizer. Routing Policy Specification Language (RPSL). *Request for Comment 2280*, Jan. 1998.

[15] T. Joachim. Text categorization with Support Vector Machines: Learning with many relevant features. *The 10th European Conference on Machine Learning*, vol.1398:pp.137–142, Springer Berlin, Apr. 2006.

[16] T. Bates, P. Smith, and G. Huston. The CIDR report. `http://bgp.potaroo.net/`.

[17] Internet Exchange Directory. `https://prefix.pch.net/applications/ixpdir/`.

[18] International Internet Service Providers. `http://www.topology.org/net/isp_int.html/`.

[19] Internet Service Providers in the USA. `http://www.topology.org/net/isp_usa.html/`.

[20] Internet Service Providers in Australia. `http://www.topology.org/net/isp.html/`.

[21] C. Chang and C. Lin. LibSVM: a Library for Support Vector Machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[22] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the MAX2SAT and MAX DI-CUT problems. *In Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*, 2002.

[23] K.Toh, M. J. Todd, and R. H. Tutuncu. A Matlab software for semidefinite-quadratic-linear programming, programming, version 4.0, Optimization Methods and Software, 11(1999), pp.545–581. `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

[24] CAIDA, AS Relationships. `http://www.caida.org/data/active/as-relationships/`.

[25] G. Siganos and M. Faloutsos. Analyzing bgp policies: Methodology and tool. *Request for Comment 2280*, Jan. 1998.