HeteroSales: Utilizing Heterogeneous Social Networks to Identify the Next Enterprise Customer

Qingbo Hu* qhu5@uic.edu
Sihong Xie* sxie6@uic.edu
Jiawei Zhang* jzhan9@uic.edu
Qiang Zhu† qzhu@linkedin.com
Songtao Guo† soguo@linkedin.com
Philip S. Yu*◇ psyu@uic.edu

*Department of Computer Science, University of Illinois at Chicago, Chicago, USA
†LinkedIn Corporation, Mountain View, USA
◇Institute for Data Science, Tsinghua University, Beijing, China

ABSTRACT

Nowadays, a modern e-commerce company may have both online sales and offline sales departments. Normally, online sales attempt to sell in small quantities to individual customers through broadcasting a large amount of emails or promotion codes, which heavily rely on the designed backend algorithms. Offline sales, on the other hand, try to sell in much larger quantities to enterprise customers through contacts initiated by sales representatives, which are more costly compared to online sales. Unlike many previous research works focusing on machine learning algorithms to support online sales, this paper introduces an approach that utilizes heterogenous social networks to improve the effectiveness of offline sales. More specifically, we propose a two-phase framework, ***HeteroSales***, which first constructs a company-to-company graph, a.k.a. ***Company Homophily Graph*** **(CHG)**, from semantics based meta-path learning, and then adopts label propagation on the graph to predict promising companies that we may successfully close an offline deal with. Based on the statistical analysis on the world's largest professional social network, LinkedIn, we demonstrate interesting discoveries showing that not all the social connections in a heterogeneous social network are useful in this task. In other words, some proper data preprocessing is essential to ensure the effectiveness of offline sales. Finally, through the experiments on LinkedIn social network data and third-party offline sales records, we demonstrate the power of HereroSales to identify potential enterprise customers in offline sales.

1. INTRODUCTION

Establishing and utilizing the social connections in online social networks like LinkedIn, Twitter, and Facebook, etc. to sell products becomes ubiquitous nowadays. This is referred to "social selling" by the general public. The success of social selling and its potential business opportunities motivate large social network companies to provide services for their sales professional members to find potential customers. For instance, the Sales Solution[1] offered by LinkedIn is a great representative of this type of services. In reality, modern e-commerce companies usually have both online and offline sales teams/departments. Online sales aim to sell a small quantity of the product to each individual customer, which are normally achieved by sending recommendation or advertising emails to users. Due to the low cost of such approach, emails can be sent in hundreds of thousands or even millions. Accordingly, obtaining the labels (whether the received users react to the emails or not) for these emails is relatively easy. Such labeled data can be used to further improve the quality of the recommendation and attract more potential customers in the future. Since online sales focus on each individual human customer, who possesses unique personal preference, researchers have found that personalized algorithms are able to better empower online sales [9, 14, 24].

Offline sales, on the other hand, focus on companies/organizations rather than individual customers. Unlike online sales, offline sales rely on the contacts established between the seller's sales agents and the representatives of the buyer, which are much more costly compared to the online sales. Sales people make phone calls or send emails to the buyer's representatives and attempt to sell a large quantity of the product in one single deal. For example, selling multiple license keys of a software to a corporation is usually achieved through offline sales. Although the offline sales take longer time and higher cost, they continue to bring in indispensable revenues and are an important part of a company's selling activities. Computer scientists seldom study how to improve the effectiveness of offline sales because of two major challenges: (1) the process of offline sales usually takes a long time and requires considerable human labor, making the data hard to collect. (2) The targets of offline sales are companies rather than real humans. In other words, companies do not have direct and meaningful "social connections" that can be exploited in this task. In this paper, we study the problem of boosting offline sales effectiveness with the assistance of online heterogeneous social network. A heterogeneous social network refers to a social network containing various types of entities and relationships. LinkedIn is a great example, and it contains different entities like companies and users, as well as various relationships like employments and social connections. With previous sales records and the help of a social network, our task is to find new companies that the seller has a high chance to close a deal.

To deal with the challenge (1), we retrieve and study nearly one-year of offline sales records, which are offered by an undisclosed

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2883000.

[1]https://business.linkedin.com/sales-solutions

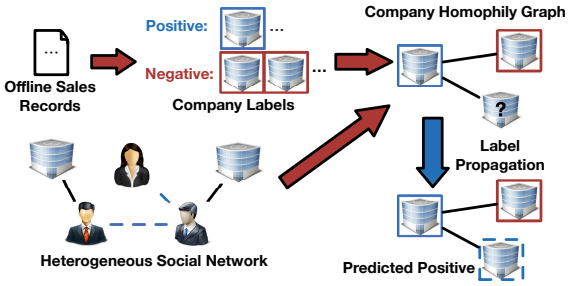


Figure 1: HeteroSales Framework. The red arrows denote the first phase of HeteroSales, and the blue arrow marks the second one.

third-party company. The offline sales records are related to a software tool that are used by the finance and sales personnels in companies. As for the challenge (2), with the help of a large heterogeneous social network (LinkedIn in our case), we utilize the social links and other relationships among companies' employees to create the connections among companies and further exploit them. More specifically, we propose a two-step framework, ***HeteroSales*** (shown in Figure 1), which first constructs a company-to-company graph, i.e. ***Company Homophily Graph*** by learning from the semantics based meta-paths on heterogeneous entities in the social network. Afterwards, *label propagation* [25, 26] will be used on the established graph to predict potential enterprise customers. Additionally, we also allow HeteroSales to incorporate the prediction results of traditional classification algorithms, such as *logistic regression* and *random forest*, to further improve its flexibility and robustness. Through statistical findings on our data set, we will show that due to potential noises, not all of the social connections in a social network are useful in the HeteroSales. The segmentation we performed on the users leads to interesting findings, such as that the social connections of the personnels who use the product are more helpful for us to discover new potential enterprise customers. Therefore, user segmentation can assist us to find those indicative social connections, while removing the noisy ones. Finally, through substantial experiments, we evaluate the proposed HeteroSales by comparing it with several baselines and demonstrate its advantages over them. We summarize our contributions as follows:

- To the best of our knowledge, this is the first paper that studies the problem of improving the effectiveness of offline sales by incorporating heterogeneous online social networks.
- We develop an innovative and general two-phase framework, HeteroSales, to address the problem of improving offline sales effectiveness. HeteroSales combines semantics based meta-path learning and label propagation to predict potential customers. Moreover, HeteroSales is also flexible enough to incorporate the prediction results of other classifiers.
- Based on substantial experiments on the data collected from real offline sales records and LinkedIn, we demonstrate that HeteroSales outperforms several baselines in predicting companies that we may potentially close offline selling deals with.
- Based on the statistical findings on the collected data set, we point out that not all the social connections in a social network are useful in predicting potential enterprise customers. We propose a user segmentation-based preprocessing that is able to filter out the noisy social connections, while unveiling additional business insights for the offline sales personnels.
- Although we only focus on improving offline sales, the HeteroSales framework is actually general enough to be used in other similar applications, such as assessing the job applicant quality with the help of heterogeneous social networks.

We organize the rest of this paper as follows: Section 2 describes the collected data set. Section 3 presents problem formulation, as well as the proposed HeteroSales framework. Section 4 has two parts: the first part focuses on the statistical analysis and user segmentation on the social network, and the second part introduces the experimental evaluation of HeteroSales. Section 5 introduces previous research work that is related to this paper. At last, Section 6 provides a conclusion for this article.

2. DATA SET DESCRIPTION

The collected data set contains two parts: offline sales records and their relevant information in a heterogeneous social network. The offline sales records are related to one product that is used by financial/selling personnels and span almost one year, which are provided by an undisclosed third-party company. The job of offline sales representatives of this product is to introduce it to other companies and try to sell multiple license keys to them. We extract the companies in the sales records, and according to the logged information, we know whether the sales representatives succeeded or failed to close a deal with the buyers. In other words, the success/failure in closing an offline deal decides the label of the company to be positive/negative. In total, we obtain 7,914 different companies. In particular, 1,467 companies are positive ones and the remaining 6,447 companies are marked as negative. Compared to the hundreds of thousands or even millions of records that we can extract from online sales, the number of offline sales records are much less. However, the win rate of the offline sales records (the ratio of positive companies, around 19%) is much higher than the average conversion rate of online sales attempts (the ratio of individuals who purchase the product/service, usually less than 5%)[2].

Next, we extract the relevant data of the 7,914 companies in the LinkedIn network. The obtained social network can help us build connections among companies that assist us to find potential enterprise customers. We first extract all the LinkedIn users who are the employees of these companies, as well as their social connections to each others. Moreover, we also retrieve a user's title, seniority level and the type of industry he/she works in. As a result, the social network we obtained from LinkedIn contains five types of heterogeneous nodes: **companies**, **users**, **titles**, **seniorities** and **industries**. Additionally, we also have five different types of links in the network: **employment links** between companies and users, **social links** between users and users, **"has-title" links** between users and titles, **"has-seniority"** links between users and seniorities, **"works-in-industry"** links between users and industries. Due to the nature of them, these five types of links are all undirected edges.

To illustrate the extracted heterogeneous network, a toy example is shown in Figure 2. There are three people, u_1, u_2 and u_3, who are employed by three companies c_1, c_2 and c_3, respectively. Besides the company and user entities, there are 3 other types of entities contained in this network: titles (square nodes), seniorities (circle nodes) and industries (polygon nodes). In this toy example, both u_1 and u_2, and u_2 and u_3 are socially connected (marked by blue dashed edges in Figure 2). Moreover, we also know that u_1 is a junior software engineer, u_2 is a senior software engineer and u_3 is a senior product manager. Therefore, u_1 and u_2 are linked to the same title node of "software engineer", while u_3 is connected

[2] http://www.smartinsights.com/ecommerce/ecommerce-analytics/ecommerce-conversion-rates/

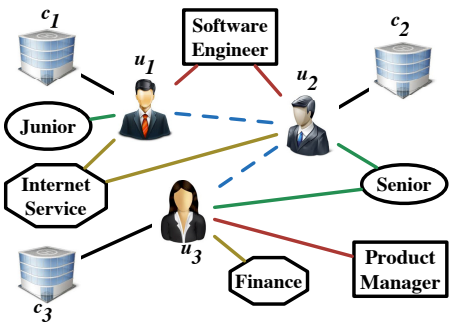


Figure 2: A Toy Example of the Heterogeneous Social Network

to the title node of "product manager". Similarly, u_2 and u_3 are connected to the seniority node of "senior", and u_1 is linked to the seniority node of "junior". Finally, user u_1 and u_2 both work in the same industry and thus they are connected to this industry node: "Internet service". User u_3, on the other hand, connects to a different industry node: "finance".

In our data set, we have retrieved 11,145,387 LinkedIn users employed by the 7,914 companies in the sales records. The huge number of extracted users is due to that there are more than 600 companies are large corporations with more than 10 thousand employees. Accordingly, we can get 147 industry nodes, 372 title nodes and 10 seniority nodes linked to the users. Besides this, the number of social connections among the users is 267,852,450.

3. PROBLEM FORMULATION AND MODEL DESCRIPTION

3.1 Notations

To begin with, we first introduce some useful notations that will be used in the later presentation. The heterogeneous social network is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ refers to the set of nodes, which contains five different types of them, i.e. $\mathcal{V} = \mathcal{C} \cup \mathcal{U} \cup \mathcal{T} \cup \mathcal{S} \cup \mathcal{D}$. $\mathcal{C} = \{c_1, c_2, ..., c_{N_\mathcal{C}}\}$ denotes the set of company nodes, c_i is the i^{th} company node and $N_\mathcal{C}$ is the total number of company nodes. Similarly, $\mathcal{U} = \{u_1, u_2, ..., u_{N_\mathcal{U}}\}$, $\mathcal{T} = \{t_1, t_2, ..., t_{N_\mathcal{T}}\}$, $\mathcal{S} = \{s_1, s_2, ..., s_{N_\mathcal{S}}\}$ and $\mathcal{D} = \{d_1, d_2, ..., d_{N_\mathcal{D}}\}$ are the sets of nodes representing users, titles, seniorities and industries, respectively, whose sizes are $N_\mathcal{U}$, $N_\mathcal{T}$, $N_\mathcal{S}$ and $N_\mathcal{D}$. $\mathcal{E}$, on the other hand, denotes set of edges in the network. As stated in the previous section, we also have five different types of edges: $\mathcal{E} = \mathcal{E}_{\mathcal{CU}} \cup \mathcal{E}_{\mathcal{UU}} \cup \mathcal{E}_{\mathcal{UT}} \cup \mathcal{E}_{\mathcal{US}} \cup \mathcal{E}_{\mathcal{UD}}$. $\mathcal{E}_{\mathcal{CU}} = \{(c_i, u_j) : c_i \in \mathcal{C} \text{ and } u_j \in \mathcal{U}\}$ denotes the set of **employment** edges between company nodes and user nodes. Similarly, $\mathcal{E}_{\mathcal{UU}} = \{(u_i, u_j) : u_i, u_j \in \mathcal{U}\}$ is the set of **social** edges among user nodes, $\mathcal{E}_{\mathcal{UT}} = \{(u_i, t_j) : u_i \in \mathcal{U} \text{ and } t_j \in \mathcal{T}\}$ is the set of **has-title** edges between user nodes and title nodes, $\mathcal{E}_{\mathcal{US}} = \{(u_i, s_j) : u_i \in \mathcal{U} \text{ and } s_j \in \mathcal{S}\}$ is the set of **has-seniority** edges between user nodes and seniority nodes, and finally, $\mathcal{E}_{\mathcal{UD}} = \{(u_i, d_j) : u_i \in \mathcal{U} \text{ and } d_j \in \mathcal{D}\}$ is the set of **works-in-industry** edges between user nodes and industry nodes. Moreover, the edges in $\mathcal{E}$ are all undirected.

Additionally, we define the company labels as $\mathcal{L} = \{l_1, l_2, ...l_{N_\mathcal{C}}\}$, where l_i is the label of c_i. We have $l_i = \{l^+, l^-\}$, where l^+ and l^- stand for "positive", "negative" labels, respectively. If the offline sales records show that we succeeded/failed to close a deal with c_i, we set $l_i = l^+/l^-$, otherwise, we consider l_i to be undefined and c_i to be an **unlabeled** company. Moreover, we also introduce additional features associated with each company, in order to facilitate traditional classifiers to provide prediction results for unlabeled companies. In the later section, we will introduce how to incorporate such results into the HeteroSales framework to further improve its robustness and flexibility. We use a vector $\mathbf{f}_i = (f_{i,1}, f_{i,2}, ..., f_{i,N_F})$ to represent the features of c_i, where $f_{i,j}$ is the value of j^{th} feature of c_i, and N_F is the total number of features. How to choose company features is independent from the proposed HeteroSales framework, which needs additional business insight and feature engineering that depends on the data set. In the experiment section, we will introduce the company features we extracted from our data set and the empirical reasons for choosing them. Given the heterogeneous social network ($\mathcal{G}$), the company features of each node ($\mathbf{f}_i$), and all the known company labels, we need to predict whether an unlabeled company has l^+ or l^-.

3.2 Meta-paths

The purpose of introducing $\mathcal{G}$ is to help us create connections between companies. The intuition is that the connected companies should be more likely to have the same label [12]. Later, we can use graph-based algorithms, such as label propagation, to exploit the graph built on such connections to infer the labels of the unlabeled companies. If a company is predicted with a high probability to be positive, it can be a good potential customer that an offline sales agent should target at. To begin with, we provide the following formal definition of the term "*meta-path*":

DEFINITION 3.1 (***Meta-path***). *A meta-path in a heterogeneous social network refers to a sequence of node types defined based on the network schema. An instance of the meta-path refers to a sequence of connected nodes in the network satisfying the node type definition of the meta-path.*

Since our task is to predict the labels of unlabeled companies, we only focus on the meta-paths connecting company nodes. We introduce the first type of meta-path that represents the direct connection between companies based on the company features:

DEFINITION 3.2 (***Feature Similarity Path***). *A feature similarity path directly connects company nodes, which is defined as $\mathcal{C}-\mathcal{C}$. The existence of a feature similarity path instance between c_i and c_j, where $c_i, c_j \in \mathcal{C}$, solely depends on their features, $\mathbf{f}_i$ and $\mathbf{f}_j$.*

Next, we also have the following meta-paths that are constructed based on the relationships in the heterogeneous social network. Since companies do not have direct links to each other in the network, all these meta-paths are constructed through the different relationships among their employees:

DEFINITION 3.3 (***Social Path***). *A social path is defined as $\mathcal{C}-\mathcal{U}-\mathcal{U}-\mathcal{C}$. An instance of social path can be denoted by $c_x - u_i - u_j - c_y$, which satisfies $c_x, c_y \in \mathcal{C}$, $u_i, u_j \in \mathcal{U}$, $(c_x, u_i), (c_y, u_j) \in \mathcal{E}_{\mathcal{CU}}$ and $(u_i, u_j) \in \mathcal{E}_{\mathcal{UU}}$. We use $\mathcal{P}^{(1)}$ to refer to the set of all social path instances.*

DEFINITION 3.4 (***Title Homophily Path***). *A title homophily path is defined as $\mathcal{C}-\mathcal{U}-\mathcal{T}-\mathcal{U}-\mathcal{C}$. An instance of title homophily path can be denoted by $c_x - u_i - t_j - u_k - c_y$, where $c_x, c_y \in \mathcal{C}$, $u_i, u_k \in \mathcal{U}$, $t_j \in \mathcal{T}$, $(c_x, u_i), (c_y, u_k) \in \mathcal{E}_{\mathcal{CU}}$ and $(u_i, t_j), (u_k, t_j) \in \mathcal{E}_{\mathcal{UT}}$. We use $\mathcal{P}^{(2)}$ to refer to the set of all title homophily path instances.*

DEFINITION 3.5 (***Seniority Homophily Path***). *A seniority homophily path is defined as $\mathcal{C}-\mathcal{U}-\mathcal{S}-\mathcal{U}-\mathcal{C}$. An instance of seniority homophily path can be denoted by $c_x - u_i - s_j - u_k - c_y$, which satisfies $c_x, c_y \in \mathcal{C}$, $u_i, u_k \in \mathcal{U}$, $s_j \in \mathcal{S}$, $(c_x, u_i), (c_y, u_k) \in \mathcal{E}_{\mathcal{CU}}$ and $(u_i, s_j), (u_k, s_j) \in \mathcal{E}_{\mathcal{US}}$. We use $\mathcal{P}^{(3)}$ to denote the set of all seniority homophily path instances.*

DEFINITION 3.6 (***Industry Homophily Path***). *An industry homophily path is defined as $\mathcal{C}-\mathcal{U}-\mathcal{D}-\mathcal{U}-\mathcal{C}$. An instance of industry homophily path can be denoted by $c_x-u_i-d_j-u_k-c_y$, which satisfies $c_x, c_y \in \mathcal{C}$, $u_i, u_k \in \mathcal{U}$, $d_j \in \mathcal{D}$, $(c_x, u_i), (c_y, u_k) \in \mathcal{E}_{\mathcal{CU}}$ and $(u_i, d_j), (u_j, d_k) \in \mathcal{E}_{\mathcal{UD}}$. We use $\mathcal{P}^{(4)}$ to refer to the set of all industry homophily path instances.*

As one may notice, the above definition does not contain second degree social connections or paths connecting more than 5 nodes. In other words, paths like $\mathcal{C}-\mathcal{U}-\mathcal{U}-\mathcal{U}-\mathcal{C}$ and $\mathcal{C}-\mathcal{U}-\mathcal{T}-\mathcal{U}-\mathcal{T}-\mathcal{U}-\mathcal{C}$ are not considered. As shown in [10, 16], such longer paths may create more noisy connections. This will further hurt the effectiveness of label propagation, the algorithm we use to predict unknown company labels. In the later experiments, we will demonstrate that how the above five defined meta-paths empower the HeteroSales framework.

3.3 HeteroSales Framework

This subsection presents the proposed HeteroSales framework. HeteroSales consists of two steps: the first step establishes a ***Company Homophily Graph*** **(CHG)** connecting companies by learning from the known company labels and the meta-paths. In the second step, HeteroSales adapts ***Label Propagation*** on the CHG to infer the unknown company labels.

3.3.1 Company Homophily Graph (CHG)

Before introducing the process of constructing Company Homophily Graph, we first provide its formal definition:

DEFINITION 3.7 (***Company Homophily Graph***). *Company Homophily Graph (CHG) is a weighted graph that only contains company nodes and undirected edges among them. Each edge is also associated with a weight between 0 to 1, which denotes the probability that the two connected companies have the same label. Formally, we denote the CHG by $\widetilde{\mathcal{G}} = (\mathcal{C}, \widetilde{\mathcal{E}}, \mathcal{W})$, where $\mathcal{C}$ is the set of company nodes, $\widetilde{\mathcal{E}} = \{(c_i, c_j) : c_i, c_j \in \mathcal{C}\}$ is the set of edges connecting companies, and $\mathcal{W}$ is a function satisfying $\mathcal{W} : (c_i, c_j) \to w_{i,j}$, $(c_i, c_j) \in \widetilde{\mathcal{E}}$, $w_{i,j} \in \mathbb{R}$ and $w_{i,j} \in [0, 1]$.*

The connectivity through the predefined meta-paths demonstrates the homophily of companies from different aspects. In other words, we can construct different CHGs based on different types of meta-paths and then fuse them through a learning task to obtain an optimal setting. This is exactly the fundamental idea of constructing the CHG in the first step of HeteroSales. We start with presenting how we construct the instances of feature similarity path only based on company features. The intuition is that if two companies have similar features, it is more likely that they have the same label, and thus they should be connected in the CHG with a larger weight. Based on such intuition, we can define the following weight between two companies, say c_i and c_j, based on their features, $\mathbf{f}_i$ and $\mathbf{f}_j$:

$$w_{i,j}^{(0)} = \exp\left[-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|^2}{\sigma^2}\right] \tag{1}$$

where $\|\cdot\|$ denotes ℓ_2 norm, and σ is a scaling factor that could be decided by cross-validation.

As one may see, Eq. (1) transforms the Euclidean distance between company features to the weight, which satisfies our previous assumption about the relationship between company features and the edge weights. Actually, similar transformation like Eq. (1) is also frequently used in many previous research work, such as [7, 25, 26]. In order to simplify the definition of $w_{i,j}^{(0)}$, if the values of different features have different scales, we adopt normalization on the feature values to ensure each feature has a mean of zero with a unit variance. This means that we let each feature equally contributes to the result computed in Eq. (1). According to Eq. (1), we end up with pairwise weight between each pair of companies. In other words, the $\widetilde{\mathcal{G}}$ defined based on Eq. (1) is a complete graph. In practice, this complete graph may not perform well, since edges with small weights do not help the inference and tend to introduce more noise. Therefore, we also need to have a threshold to cut off those edges having weights smaller than it. Similar to σ, the value of the threshold is selected by cross-validation. After removing noisy edges, we denote the obtained CHG as $\widetilde{\mathcal{G}}^{(0)}$.

Next, for other meta-paths, we use PathSim [16] to define the weight between two companies. PathSim is a well developed weighting technique to measure the strength of the connectivity of two nodes through meta-path, and it was widely used to solve other problems related to heterogeneous social networks [16,21,22]. PathSim defines the weight between two companies, say c_i and c_j, as:

$$w_{i,j}^{(k)} = \frac{2 \times |\{p_{i,j} : p_{i,j} \in \mathcal{P}^{(k)}\}|}{|\{p_{i,i} : p_{i,i} \in \mathcal{P}^{(k)}\}| + |\{p_{j,j} : p_{j,j} \in \mathcal{P}^{(k)}\}|} \tag{2}$$
$$where\ k = \{1, 2, 3, 4\}$$

In Eq. (2), $\{p_{x,y} : p_{x,y} \in \mathcal{P}^{(k)}\}$ denotes the set of instances of $\mathcal{P}^{(k)}$, of which the company nodes at two end points are c_x and c_y. $|\cdot|$ refers to the total number of instances in a set. The numerator of Eq. (2) is the strength of connectivity between c_i and c_j, denoted by the count of unique meta-path instances between c_i and c_j in terms of both directions. The denominator of Eq. (2) is the aggregated visibility of c_i and c_j, which is defined as the summation of the number of meta-path instances between c_i to itself and the number of instances between c_j to itself. Roughly speaking, if two companies c_i and c_j are connected through more meta-path instances, $w_{i,j}^{(k)}$ tends to get larger. Otherwise, $w_{i,j}^{(k)}$ becomes closer to 0, indicating c_i and c_j do not connect well. If $w_{i,j}^{(k)} = 0$, it means that there is no edge connecting c_i and c_j in the constructed CHG. Accordingly, we denote the CHG defined based on $\mathcal{P}^{(k)}$ ($k = 1 \sim 4$) under Eq. (2) by $\widetilde{\mathcal{G}}^{(k)}$.

Notice that in order to ensure Eq. (2) is properly defined and has a value between 0 and 1, we need to guarantee that there is at least one $p_{i,i}$ satisfying $p_{i,i} \in \mathcal{P}^{(k)}$. Similar constraint also applies on $p_{j,j}$. However, the data retrieved from a social network usually does not include a social connection between each user to himself/herself, which causes problems for $w_{i,j}^{(1)}$, i.e. the weight defined for the social path. Considering a heterogeneous social network that only contains one social path instance $c_i - u_x - u_y - c_j$, we will have the count of social paths between c_i and c_j to be 1, but the count of social paths between c_i to itself and c_j to itself are both 0, making Eq. (2) not well defined. To solve that, we just need to manually add a social connection between each user to himself/herself. In the previous example, $w_{i,j}$ will have the value of 1 after such correction, which is exactly the maximal value we want $w_{i,j}$ to take, since there is only one social path instance $c_i - u_x - u_y - c_j$ in the graph. This augmentation added on social connections to ensure a well defined $w_{i,j}^{(1)}$ does not need to be applied to $w_{i,j}^{(2)}$, $w_{i,j}^{(3)}$ and $w_{i,j}^{(4)}$. The reason is that a user always shares the same title/seniority/industry with himself/herself. For example, a title homophily path instance between c_i to itself, like $c_i - u_x - t_k - u_y - c_i$, always exists, since u_x and u_y can be the same user. Unlike $\widetilde{\mathcal{G}}^{(0)}$, not every pair of companies are connected through meta-path instances in the social network, which results in that $\widetilde{\mathcal{G}}^{(1)}$, $\widetilde{\mathcal{G}}^{(2)}$, $\widetilde{\mathcal{G}}^{(3)}$ and $\widetilde{\mathcal{G}}^{(4)}$ are not complete graphs. Therefore, the process of removing noisy edges to control the graph density does not need to be used in these CHGs.

At last, we introduce the learning task to fuse $\widetilde{\mathcal{G}}^{(0)} \sim \widetilde{\mathcal{G}}^{(4)}$ into a unified CHG based on the observed company labels from the offline sales records. Since each CHG provides a unique perspective to indicate the closeness of two companies, we use a linear combination of $w_{i,j}^{(0)} \sim w_{i,j}^{(4)}$ to define the $w_{i,j}$ in the final fused CHG. However, since $w_{i,j}$ has a constraint that $0 \leq w_{i,j} \leq 1$, we apply the sigmoid function on the linear combination, which yields:

$$w_{i,j} = \frac{1}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)} \tag{3}$$

where $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)^\top$, α_k ($\alpha_k \in \mathbb{R}$) is the linear weight associated with $w_{i,j}^{(k)}$, $\mathbf{w}_{i,j} = \left(w_{i,j}^{(0)}, w_{i,j}^{(1)}, w_{i,j}^{(2)}, w_{i,j}^{(3)}, w_{i,j}^{(4)}\right)^\top$, and ϵ ($\epsilon \in \mathbb{R}$) is a residual term. Remember that $w_{i,j}$ is defined to be the probability that c_i and c_j have the same label, we thus can learn each α_k and ϵ through ***maximum likelihood estimation*** (MLE) based on the observed company labels. In particular, the negative log-likelihood of the observed data, denoted by ℓ, contains two parts: the first part is the likelihood that all pairs of observed companies having the same label, and the second part is the likelihood that all pairs of observed companies having different labels. If we denote the set of observed company pairs having the same label by $U_1 = \{(i,j) : i \neq j, l_i = l_j\}$, and let $U_2 = \{(x,y) : l_x \neq l_y\}$ be the set of all pairs of observed companies with different labels. Mathematically, ℓ can be computed as:

$$\begin{aligned}
\ell &= -\left[\sum_{(i,j)\in U_1} \log w_{i,j} + \sum_{(x,y)\in U_2} \log\left(1 - w_{x,y}\right)\right] \\
&= \sum_{(i,j)\in U_1} \log\left[1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)\right] + \\
&\quad \sum_{(x,y)\in U_2} \left\{\log\left[1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{x,y} + \epsilon\right)\right] + \left(\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{x,y} - \epsilon\right)\right\} \\
&= \sum_{i\neq j} \log\left[1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)\right] + \sum_{(x,y)\in U_2} \left(\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{x,y} - \epsilon\right)
\end{aligned}$$

Accordingly, the MLE problem we need to solve is:

$$\text{argmin}_{\boldsymbol{\alpha},\epsilon}\, \ell$$

We use *gradient descent* method with line search [3, 4, 11, 13] to solve this problem. Starting with the initial values of $\boldsymbol{\alpha}$ and ϵ, which are randomly generated, the gradient descent method will iteratively use line search to find better values of $\boldsymbol{\alpha}$ and ϵ to further minimize ℓ, until a local optimum is achieved. The proof to show that the gradient descent will converge can be found in [3,4,11,13], so we do not present it in this article, due to the space limit. The line search follows the gradients of each α_k and ϵ, which are readily computed:

$$\begin{aligned}
\frac{\partial \ell}{\partial \alpha_k} &= \sum_{i\neq j} \frac{\exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right) \cdot (-w_{i,j}^{(k)})}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)} + \sum_{(x,y)\in U_2} w_{x,y}^{(k)} \\
&= \sum_{i\neq j} \left[1 - \frac{1}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)}\right] \cdot (-w_{i,j}^{(k)}) + \sum_{(x,y)\in U_2} w_{x,y}^{(k)} \\
&= \sum_{i\neq j} \frac{w_{i,j}^{(k)}}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)} - \sum_{(x,y)\in U_1} w_{x,y}^{(k)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ell}{\partial \epsilon} &= \sum_{i\neq j} \frac{\exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)} - |U_2| \\
&= \sum_{i\neq j} \left[1 - \frac{1}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)}\right] - |U_2| \\
&= |U_1| - \sum_{i\neq j} \frac{1}{1 + \exp\left(-\boldsymbol{\alpha}^\top \cdot \mathbf{w}_{i,j} + \epsilon\right)}
\end{aligned}$$

where $|U_1|$ and $|U_2|$ are two constant factors, denoting the numbers of pairs of observed companies having the same label and different labels, respectively. After we learned the optimal $\boldsymbol{\alpha}$ and ϵ, we can use Eq. (3) to obtain the weights between all companies in the social network and construct the final CHG. Of course, if $i = j$ or $w_{i,j} = 0$, there will be no edge connecting c_i and c_j in the CHG.

3.3.2 Label Propagation on the Fused CHG

The second phase of HeteroSales is to use ***label propagation*** [25, 26] on the constructed CHG to infer the labels for unlabeled companies. Generally speaking, starting from the known company labels, the label propagation allows each node to iteratively propagate a message to its neighbors, and the message is constructed based on the node's belief (will be defined in Def. 3.8 later), as well as the edge weights connecting it with the neighbors. Afterwards, each node will update its belief by normalizing the messages it receives. The iterative process of sending message and updating believes continues until the believes of nodes converge.

In order to introduce the second phase of HeteroSales in details, we first provide the definition of the belief of a company node.

DEFINITION 3.8 (***Belief***). *The belief of a company node is the probability that this company has a l^+ label. Mathematically, we use b_i to denote the belief of c_i, which should satisfy: $b_i \in \mathbb{R}$ and $b_i \in [0, 1]$.*

According to the definition of belief, we can immediately draw the following conclusion: $l_i = l^+ \rightarrow b_i = 1$ and $l_j = l^- \rightarrow b_j = 0$. In other words, if we already observe the label of a company node, its belief will be fixed. Therefore, label propagation aims to obtain a stable belief for each unlabeled company node, which indicates whether the company should be recommended to the offline sales representatives. Let $b_i^{(k)}$ be the computed belief of c_i in the k^{th} iteration, then the label propagation used in the HeteroSales can be presented in the following steps:

(i) Initialization. Set the initial believes of company nodes as:

$$b_i^{(0)} = \begin{cases} 1 & l_i = l^+ \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

(ii) Message passing. In the k^{th} iteration, for each c_i, it constructs the following message to send to its each connected neighbor in the CHG:

$$m_{i\rightarrow j}^{(k)} = w_{i,j} \cdot b_i^{(k)}, \forall j \text{ that } (c_i, c_j) \in \widetilde{\mathcal{E}}$$

(iii) Belief updating. After receiving the messages obtained from all neighbors, c_i can updates its own belief:

$$b_i^{(k+1)} = \frac{\sum_{(c_j,c_i)\in\widetilde{\mathcal{E}}} m_{j\rightarrow i}^{(k)}}{\sum_{(j,i)\in\widetilde{\mathcal{E}}} w_{j,i}} = \frac{\sum_{(j,i)\in\widetilde{\mathcal{E}}} w_{j,i} \cdot b_j^{(k)}}{\sum_{(c_j,c_i)\in\widetilde{\mathcal{E}}} w_{j,i}}$$

(iv) Belief resetting. The believes of companies with known labels must be reset, since they are already observed:

$$b_i^{(k+1)} = \begin{cases} 1 & l_i = l^+ \\ 0 & l_i = l^- \end{cases} \tag{5}$$

(v) Repeating. We iteratively run steps (ii)∼(iv) until all believes converge.

It is provable that the label propagation algorithm described in the above five steps guarantees the believes will converge. Due to the space limit, we will not illustrate the proof process, which can be found in [25,26]. Moreover, the work in [25,26] also shows that a closed-form solution for the believes exists. However, we found the closed-form solution is not efficient when applied to a huge data set that may contain millions of unlabeled nodes, which is a frequent scenario we may encounter in practice. Moreover, similar to what is introduced in [7], the closed-form solution involves the calculation of the inverse of a matrix, which may cause additional problems if singular points exist. Alternatively, it is usually much easier to directly implement the iterative label propagation in the above steps (i)∼(v) on a distributed system to deal with a large data set with millions of unlabeled nodes. For example, we can directly implement steps (i)∼(v) in the GraphX of Apache Spark[3]. Therefore, in our experiments, we also implement an iterative version of the label propagation to assist us to report important factors, such as the number of iterations required for the believes to converge. Information like this number may be of great interest to the readers who intend to implement it in a distributed system, as it affects the efficiency of the algorithm.

Next, we discuss about the details of our implementation of the label propagation. Let $\mathbf{b}^{(k)} = \left(b_1^{(k)}, b_2^{(k)}, ...b_{N_\mathcal{C}}^{(k)}\right)^\top$ and W denote the matrix that contains all $w_{i,j}$:

$$W = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{N_\mathcal{U},1} \\ w_{1,2} & w_{2,2} & \dots & w_{N_\mathcal{U},2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,N_\mathcal{U}} & w_{2,N_\mathcal{U}} & \dots & w_{N_\mathcal{U},N_\mathcal{U}} \end{bmatrix}$$

Notice that W is a symmetric matrix, i.e. $w_{i,j} = w_{j,i}$, according to Eq. (3). We first row-wise normalize W: divide each element in W by the summation of the values of all the elements in the same row. If we use W' to denote the matrix obtained by row-wise normalizing W, steps (ii) ∼ (iii) in the label propagation algorithm is actually equivalent to:

$$\mathbf{b}^{(k+1)} = W' \cdot \mathbf{b}^{(k)} \tag{6}$$

Finally, our implementation of the label propagation can be summarized in Algorithm 1.

Algorithm 1 Label Propagation in HeteroSales

Input: W: weights in the CHG
Output: $\mathbf{b}$: believes of the companies
1: Initialize $\mathbf{b}$ according to Eq. (4)
2: Calculate W' by row-wise normalizing W
3: **while** ($\mathbf{b}$ is not converged) **do**
4: Update $\mathbf{b}$ according to Eq. (6)
5: Reset the believes of labeled nodes according to Eq. (5)
6: **end while**
7: return $\mathbf{b}$

3.3.3 Improvement of HeteroSales

In the following paragraphs, we introduce the additional steps to be added to the proposed HeteroSales framework to increase its flexibility and robustness.

Pruning on the CHG. In practice, it is possible that Eq. (3) leads to a fused CHG with many edges of small valued weights. This increases the density and the noise in the CHG, which affects the performance of the label propagation in the second phase of the HeteroSales. Therefore, instead of using $w_{i,j} = 0$ as the condition of no edge connecting c_i and c_j in the CHG, we may use a threshold with a value larger than 0 to determine whether the edge exists or not. Practically, we use $(1 + e^\epsilon)^{-1}$ as the threshold, i.e. the weight value when there is no edge between c_i and c_j in any $\widetilde{\mathcal{G}}^{(k)}$, $k = 0, ..., 4$. We set $w_{i,j} = 0$ and remove the corresponding edge, if $w_{i,j} \leq (1 + e^\epsilon)^{-1}$.

Smoothing on the CHG. As introduced in [25, 26], adopting a smoothing process on the CHG can increase the robustness of the label propagation. Therefore, we add a smoothing factor, which is a very small number to the $w_{i,j}$ between every pair of c_i and c_j. In practice, we select the value of the minimal non-zero weight in the CHG divided by 10^4 as the smoothing factor.

Combining with external classifiers. The HeteroSales framework is also flexible enough to incorporate the results of traditional classifiers, such as random forest and logistic regression. We first train the classifiers based on the company features and predict the probability that a company node has l^+ label. Suppose $\mathbf{r}_i = (r_{i1}, r_{i2}, \dots, r_{iN_\mathcal{C}})^\top$ represent the obtained probabilities from the i^{th} classifier, where r_{ij} is the probability that $l_j = l^+$ under the prediction of the i^{th} classifier, then we change the updating formula in Eq. (6) to:

$$\mathbf{b}^{(k+1)} = \beta_0 \cdot W' \cdot \mathbf{b}^{(k)} + \sum_{i>0} \beta_i \cdot \mathbf{r}_i \tag{7}$$

where β_0 is the weight associated with the updated believes from the label propagation, and $\beta_i (i > 0)$ is the weight associated with the results from the i^{th} classifier. Of course, we have the constraint $\sum_{i=0} \beta_i = 1$ to ensure the updated believes are still in the interval of $[0, 1]$. One remark related to Eq. (7) is that if we set $\beta_0 = 0$, the HeteroSales framework will be equivalent to the ensemble of traditional classifiers. In practice, we use cross validation to adjust the value of each β_i. Through experiments, we found that the optimal value of each β_i may vary along with the change of the ratio of unlabeled nodes. In other words, the prediction power of the original HeteroSales and other traditional classifiers may be different in different problem settings. Therefore, incorporating the results of external classifiers not only adds more flexibility to the HeteroSales framework, but also improves its robustness and performance. We will introduce this with more details in the next section.

4. EXPERIMENTS

This section introduces the statistical analysis and experimental evaluation we performed on the proposed HeteroSales framework. We first introduce the statistical findings to show that although we may have a large number of social connections in a social network, not all of them are useful for the HeteroSales. Afterwards, we will evaluate the HeteroSales through experiments on our data set and compare the results with several baselines.

4.1 Social Network User Segmentation

As we mentioned in the section of introducing the data set, we have around 8 thousand company nodes. The number of user nodes who are employees of these companies reaches more than 11 million and the social connections among them are around 267 million. Considering all these users and social connections in the network to construct the CHG might add potential noise. More particularly, if we include many social connections linking two users who work for companies of different labels, we may end up establishing many edges connecting two companies with different labels in the CHG. This affects the performance of the label propagation, which is largely dependent on the quality of the edges in

[3]http://spark.apache.org/graphx/

Titles	$mean_p$	$mean_n$	p-value
Software Developer	0.141	0.157	8.79e-235
Personal Banker	0.165	0.121	6.3e-212
Salesperson	0.149	0.136	2.98e-178
Mortgage/Loan Officer	0.143	0.097	2.75e-82
Manufacturing/Mechanical Engineer	0.136	0.152	2.04e-73

Table 1: Top 5 titles with the smallest p-values in the t-test

the CHG. As a matter of fact, through initial experiments, if we include all the social connections, we found that the HeteroSales is even worse than traditional classifiers, such as logistic regression and random forest. Next, we will show that only the users with certain titles/seniorities/industries can be useful in the HeteroSales.

According to the labels of the companies that users work for, we split users into two groups: positive group and negative group. Afterwards, for each user in each group, we extract all the socially connected users who work for different companies rather than his/her own company. Based on the extracted neighbors for each user, we compute the ratio of them who work for positive companies. Next, we add filters on both of the two groups to extract the users of certain titles/seniorities/industries, as well as the calculated ratio values associated with them. We expect that if two users connected in the network link to companies of the same label, the obtained CHG will have less noisy meta-path instances connecting companies of different labels. Therefore, we want to find those title/seniority/industry filters to ensure the ratio values in the positive group should be generally larger than the values in the negative group. Accordingly, after adding each filter on the two groups, we use t-test to check the difference of the ratio value distribution of the users in the positive group and negative group. More specifically, since the positive and negative companies are not the same, the number of users in the two groups can be imbalanced as well. As a result, we use Welch's t-test [19] in this task, as it takes such imbalance into consideration. If we use $mean_p$ to denote the mean of the ratio values in the positive group, and use $mean_n$ to denote the mean of the ratio values in the negative group, we list 5 titles with the most significance level, i.e. the smallest p-values in the t-test, in Table 1.

As readers may find out in Table 1, two of the five titles, i.e. software developer and manufacturing/mechanical engineer, actually have $mean_p < mean_n$. This tells us that although the ratio values of the users with these titles in the positive group and negative group are statistically significantly different, it is not helpful to consider them in the CHG construction phase. The reason behind this is that the users of these titles in positive companies are even more likely to connect to users linking to negative companies, comparing to the users in the negative companies. In other words, it is more preferable to select the users with titles that ensure $mean_p > mean_n$, which will have more meta-path instances connecting two companies of the same label. Another interesting finding in Table 1 is that we can see the similarities among titles in this test. For example, we found that engineering background users are more likely to have $mean_p < mean_n$, while finance or sales background users are more likely to have $mean_p > mean_n$. By considering that the product is used by the finance or sales personnels, this finding suggests that the social connections of the users who are more familiar with the product are more likely to lead us to new enterprise customers.

Similar discoveries can also be found when we adopt the same t-test on the industries of the users, where users belonging to the finance or sales related industries are more likely to have $mean_p > mean_n$. The results on the seniorities of the users, on the other hand, show that users with higher seniorities, such as manager-level and senior-level personnels, will be more indicative in the prediction task. Although the preferable titles/seniorities/industries are likely to change when the product in the offline sales records changes, the process to obtain them is general enough to be used in other data sets. Thus, Welch's t-test assists us to find the users with certain titles/seniorities/industries that are helpful for the HeteroSales, while removing the users with certain titles/seniorities/industries that may hurt its performance. Moreover, the same process can also help us discover additional valuable product-related business insights. In the later experiments to evaluate the HeteroSales, we only keep the users with the 10 titles, 2 seniorities and 10 industries having the most significance level in the t-test, while also ensuring $mean_p > mean_n$.

Company Feature	Meaning
#LinkedIn users	Number of LinkedIn users working for the company
#Finance/sales personnels	Number of the company's finance/sales personnels in LinkedIn
Social Selling Index[4]	A metric measuring the sales activities on social networks
#LinkedIn connections	Total number of social connections of the employees in LinkedIn
#Connections to the seller	The number connections of between the employees and the employees of the selling company in LinkedIn
#Non-employee followers	Number of non-employee followers of the company's LinkedIn page
Sales expense	US dollars spent in sales in the last year

Table 2: Company Features

4.2 Evaluation of HeteroSales

We list all the extracted company features in our experiments in Table 2. As we previously introduced, since the product is used by finance/sales personnels, we have extracted many product dependent company features, i.e. features related to a company's finance/sales professionals. To be more specific, #LinkedIn members and #finance/sales personnels are extracted to get an approximate evaluation of the company's size and its finance/sales personnels. Social Selling Index, #LinkedIn connections and sales expense estimate the active level of a company's selling activities. #Non-employee followers reflects the popularity of the companies, and #connections to the seller is a direct measure to show the connectivity between the company and the third-party company that provides us the offline sales records. As one may see, the features extracted in our experiments are based on empirical reasons, as well as the data source. The best company features used in the HeteroSales are likely to be different in other data sets, which requires additional business insights and knowledge of the products in the offline sales records. Next, we compare the performance of the following methods in the experiments:

Logistic Regression. Logistic regression [5] is a traditional classification algorithm we adopted in the experiments due to its wide usage and flexibility. We directly apply logistic regression on the company features introduced in Table 2 to get the prediction results. Moreover, we also incorporate the prediction results from logistic regression into the HereroSales framework.

Random Forest. Unlike traditional decision tree algorithm, random forest [1, 2] is an ensemble algorithm based on multiple decision trees and each of them is generated on a subset of the features. Since random forest is able to correct the overfitting problem introduced by normal decision tree, it is more robust and also frequently used in real life. Similar to the logistic regression, we directly adopt

[4] https://business.linkedin.com/sales-solutions/social-selling/the-social-selling-index

random forest on the company features, and its prediction results are also incorporated in the HeteroSales framework.

HeteroSales on Meta-path (HS-MP). This is the method we introduced in the previous sections, which contains two phases: CHG construction and label propagation. As we presented previously, in order to improve the model's robustness, we apply pruning and smoothing process on the CHG, as well as incorporate the predictions of the logistic regression and random forest as the external classifiers' results.

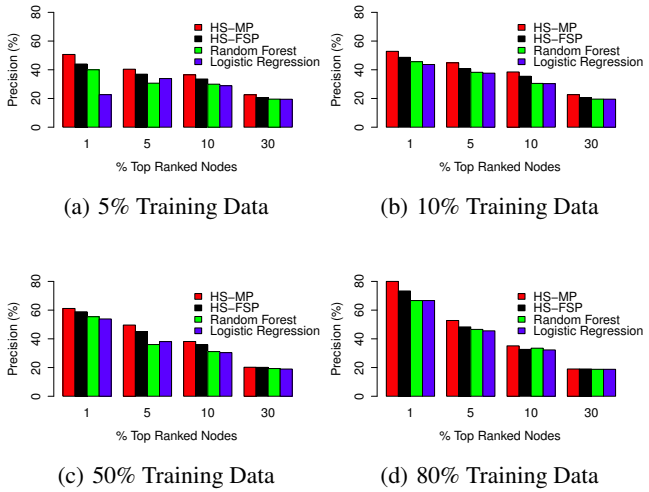
HeteroSales on Feature Similarity Path (HS-FSP). We also evaluate the HeteroSales framework only on the feature similarity path. We add this baseline since we only feed company features to the traditional classification algorithms, i.e. logistic regression and random forest, it is reasonable to check how the HeteroSales performs with the exactly same features. In other words, instead of creating a fused CHG based on learning on different meta-paths, we just need to directly apply label propagation to $\widetilde{\mathcal{G}}^{(0)}$, which is the CHG created only based on feature similarity path. Since $\widetilde{\mathcal{G}}^{(0)}$ is the result obtained after pruning and we do not have the learned $\boldsymbol{\alpha}$, we will not apply the pruning step for HS-MP in this case. However, we still add the smoothing and incorporate the prediction results from logistic regression and random forest, in order to create a fair comparison with HS-MP.

In the experiments, we first divide the companies into a training set and a testing set. Correspondingly, the company nodes in the training set will have positive and negative labels, while the company nodes in the testing set will be treated as unlabeled. By varying the ratio of training and testing sets, we use each algorithm to attempt to predict the unknown company labels from the known ones in different experimental settings. In particular, we randomly sample 5%, 10%, 50% and 80% from all companies to form the training set, and use the remaining 95%, 90%, 50% and 20% of the companies as the testing set. In these four different settings, each algorithm will output a real number in the interval of [0, 1] for each unlabeled company node, which indicates the probability for this node to have a positive label. By comparing these results with the ground truth labels, we evaluate different algorithms in three metrics: area under the curve (AUC), the precision of top recommendations and the precision vs. recall (PVR) curve. In order to conduct a fair comparison, all the algorithms are developed in R language and run in the same machine.

We first present the results of the AUC. The AUC refers to the area under the receiver operating characteristic (ROC) curve, which should be a value in the interval of [0, 1]. In general, the prediction results of a better algorithm should yield a larger value of AUC. We list the AUCs of different algorithms in the four experimental settings in Table 3. As we can see, on the one hand, the AUCs of the two HeteroSales algorithms are constantly better than the two traditional classification algorithms. On the other hand, by comparing the two HeteroSales methods, HS-MP always outperforms HS-FSP, which means the additional meta-paths based on the relationships in the heterogeneous social network bring more useful information to the HeteroSales framework. Moreover, we also see that the HS-MP has a more obvious advantage over traditional classification algorithms when the ratio of training samples is small. For example, in the scenario of training ratios to be 5% and 10%, the AUCs of HS-MP are around 4%∼5% larger than the AUCs of the random forest. In other words, we can see that the HeteroSales framework is especially useful in the case when we only have a limited number of training data. As we introduced in the beginning of this paper, it is usually very expensive to obtain the offline sales records, which require much intensive human labor. Therefore, in real life, we are often dealing with a situation that we may only

Model \ Training Ratio	5%	10%	50%	80%
HS-MP	0.665	0.674	0.697	0.732
HS-FSP	0.624	0.645	0.689	0.704
Random Forest	0.612	0.632	0.668	0.691
Logistic Regression	0.608	0.619	0.642	0.658

Table 3: AUC comparison of different algorithms



(a) 5% Training Data (b) 10% Training Data

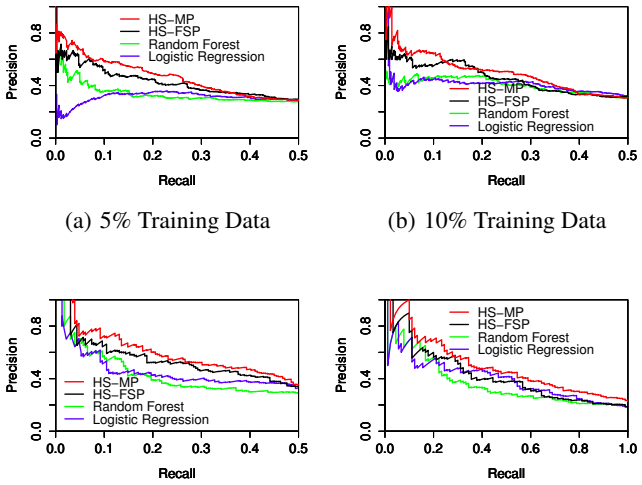
(c) 50% Training Data (d) 80% Training Data

Figure 3: Precision of Top Ranked Nodes

have a handful of labeled companies (training data), while the pool of future potential companies (testing data) can be much larger. For example, HeroSales can be very powerful in the scenario where a new product/service is available for offline sales and limited transaction data can be leveraged.

The results shown in the precision of top recommendations will demonstrate a more clear practical advantage of the HeteroSales framework over the traditional algorithms. In these experiments, for each experimental setting, we rank all the unlabeled nodes in a descending order based on the their scores obtained from each algorithm, then we compute the percentage of positive companies among the top $k\%$ (where k varies in the experiments) ranked nodes, i.e. the precision of the top $k\%$ ranked nodes. We especially set up this experiment, since the offline sales are more costly and their resources are usually limited in real-life. Therefore, we should lead the offline sales to focus on the most promising enterprise customers. In other words, offline sales agents would usually only have the chance to outreach the very top recommendations in a company list that sorted based on the probabilities they can close a deal with.

We demonstrate the precisions of top ranked companies of different algorithms under different experimental settings in Figure 3. As shown in Figure 3, the precision of top 1%∼10% ranked companies of the HS-MP clearly dominates the other algorithms under any different settings. Such dominance is especially obvious on the very top ranked companies: for the top 1% and 5% ranked companies, HS-MP can have around $6\% \sim 15\%$ higher precision comparing to the traditional classification algorithms. Although HS-FSP also generally outperforms the traditional algorithms, we still see that it is not as good as HS-MP in most cases. This further illustrates the advantage of adopting additional meta-paths in the HeteroSales framework. The precision of the top 30% ranked companies of all the algorithms is very similar in different settings. Taking a closer



(a) 5% Training Data (b) 10% Training Data

(c) 50% Training Data (d) 80% Training Data

Figure 4: Precision vs. Recall

look at it, we can see that the precision values of top 30% ranked companies are all close to 19%, which is exactly the precision of a random classifier can achieve, since the ratio of positive companies in all companies is $1467/7914 \approx 0.19$. This means that it is generally hard for an algorithm to obtain obvious lift comparing to a random classifier, if we need to recommend more companies. However, as we stated in the previous paragraph, due to the limited human resource of offline sales personnels, the quality of the top recommended companies is more important in this task, where the HS-MP appears to be very successful.

Next, we draw the precision vs. recall (PVR) curves of different algorithms under different experimental settings in Figure 4. In each of these four figures, the x-axis is the recall, while the y-axis represents the precision. When the ratios of training samples are 5%, 10% and 50%, since all the algorithms will have very similar precision values when the recall values get larger than 0.5, the PVR curves of different algorithms become indistinguishable in the recall range of (0.5, 1]. Therefore, we only draw the figure in the recall range of [0, 0.5] in Figure 4(a), Figure 4(b) and Figure 4(c). Generally speaking, the two curves (the red and black ones) representing the two HeteroSales algorithms are always above the other two curves. In other words, when we have a fixed recall value, the prediction results generated by these two HeteroSales methods usually have higher precision than the results obtained from random forest and logistic regression, no matter what the percentage of companies we use to train the models. Moreover, by comparing HS-MP to HS-FSP, we can conclude that HS-MP is also better than the HS-FSP, since HS-MP's PVR curve (red one) is above the HS-FSP's PVR curve (black one) almost everywhere in Figure 4. This further confirms the necessity of using different meta-paths in the HeteroSales framework, rather than only using company feature based meta-path, i.e. feature similarity path.

At last, we provide more supplementary information and intriguing findings we noticed in the experiments. When we run the HS-MP algorithm, although the learned values of $\alpha_0 \sim \alpha_4$ change under different experimental settings, the inequality of $\alpha_1 > \alpha_0 > \alpha_4 > \alpha_3 > \alpha_2$ always holds. This means that no matter how we change the proportion of samples used as training data, the first phase of HS-MP leads to a stable conclusion that the importance of different meta-paths are ordered as: social path > feature similarity path > industry homophily path > seniority homophily path > title homophily path. It seems that after filtering noisy social connections through the user segmentation, the remaining social connections in the network can be very indicative to reveal the potential enterprise customers. The extracted company features, on the other hand, are also very important, which explains why the traditional classification algorithms based on them are able to catch up the performance of HeteroSales when we increase the size of training data. Besides this finding, when we incorporate random forest and logistic regression into the HetereoSales framework, β_0, β_1 and β_2 (i.e. the weights associated to the results of label propagation, random forest and logistic regression, respectively) selected by cross validation are likely to change when the size of training data varies. In general, β_0 has larger values when we have less training data. The reason behind this is that as we can see in the previously introduced experimental results, traditional classifiers have worse performance when the training data is limited. The belief scores obtained from the label propagation, on the other hand, is more powerful, and thus we should accordingly trust more on them, which yields a larger value of β_0 in this case. By only comparing between the two traditional classification algorithms, random forest generally outperforms the logistic regression, and thus we always have $\beta_1/\beta_2 \approx 3:2$, accordingly. At last, the recorded number of iterations for the HS-MP algorithm to converge is different when the ratio of testing data changes. In general, when we have less testing data, HS-MP will converge more rapidly. In the experiments, we observe that the HS-MP requires around 22$\sim$107 iterations to converge when the ratio of testing data is 20%$\sim$95%.

5. RELATED WORK

The first category of related research of this paper is meta-path learning. Learning from meta-path in heterogeneous social networks can be used to solve different problems in real life [8, 17, 21–23]. [23] utilizes the meta-paths defined on multiple social networks to predict missing links among users. [22] and [21] attempt to provide more accurate online recommendations on objects such as movies/check-in locations by learning from the meta-paths discovered in heterogeneous information networks. Besides the difference of the learning model we used on the meta-paths, the application of our work is also very different from these articles. Research like [21–23] utilizes meta-path learning to solve problems within the online world. Our work, on the other hand, attempts to improve the effectiveness of an activity existed in the offline world, i.e. offline selling process, which brings us unique requirements. For example, due to the offline resource is limited, we are especially interested in the quality of top ranked recommendations in the experimental evaluation. The closest works related to ours are [15, 16]. We use the PathSim introduced in [16] to define the weight on the CHG, and the model introduced in [15] also uses a sigmoid-based function to link the meta-paths to the likelihood of data samples. However, unlike [15, 16], the HeteroSales framework is a two-step framework, which uses an additional label propagation step to generate the prediction results.

Another type of related work belongs to the label propagation algorithm. More specifically, we adapt the method in [25, 26] into the HeteroSales framework. [25] directly applies the label propagation on a predefined network, yet we use a much more complicated learning task based on meta-paths to create the network. Moreover, unlike the work in [26], we do not define the network as a random Markov field, which requires additional definition on the node and edge functions. [6, 7] adopt label propagation to predict the value of

an online video in a video network connected based on their content. However, [6, 7] also require the definition on the node and edge functions. Moreover, the HeteroSales framework also incorporates the results of external classifiers to the label propagation, which further improves its flexibility and robustness. In [18, 20], the authors use a propagation-based method to estimate product quality on a heterogeneous network. However, they construct the graph from raw data, instead of combining multiple graphs.

Finally, we also find several related articles [9,14,24] focusing on improving the effectiveness of selling process. For example, [24] studies the problem how to combine several products into one bundle and recommend it to customers through emails. Since discounts are usually also applied to these product bundles, users may more likely to purchase them, if the bundle contains different product he/she may need. However, all these research works focus on online sales process, which require a large number of training data to fully unleash the power of their proposed methods.

6. CONCLUSION

In reality, online and offline sales are the two important parts in the modern selling activities. Unlike online sales selling a small quantity of a product to each customer through methods like sending recommendation emails, offline sales normally target at enterprise customers to sell a large quantity of the product in one single deal. Since the offline sales usually require human contact between the sales agents and the representatives of the buying company, it is more costly and requires longer time and additional human labor to collect the data to be used to research on this problem. Therefore, how to find a new enterprise customer usually depends on the experience and insights of the sales personnels, and computer scientists have rarely get involved in this problem. In this paper, we have introduced a method to utilize the information in an online heterogeneous social network to improve the effectiveness of offline sales. We propose a two-step framework, HeteroSales, to achieve this goal. The HeteroSales first constructs a Company Homophily Graph (CHG) through learning from semantics based meta-paths in the social network, and then adopts a label propagation algorithm on it to find new potential enterprise customers. Based on the offline sales records of a third-party company and a large professional social network, LinkedIn, we introduce statistical findings to show that not all the users' social connections in a network can be helpful for the HeteroSales. For example, we found that those people who are more familiar with the product are more likely to have indicative social connections to help us find new enterprise customers. Finally, based on the extracted data set, we conduct extensive experiments to evaluate the proposed HeteroSales, and show it can constantly outperform other baselines in this task.

7. ACKNOWLEDGEMENT

This work is supported in part by LinkedIn Corporation, NSF through grants III-1526499, CNS-1115234, and OISE-1129076, and Google Research Award.

8. REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, 2001.

[2] L. Breiman. Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, 2002.

[3] L. Clark, D. Pregibon, J. Chambers, and T. Hastie. Tree-based models. *Statistical models in S*, 1992.

[4] A. J. Dobson and A. Barnett. *An introduction to generalized linear models*. Chapman and Hall, 1990.

[5] D. A. Freedman. *Statistical models: theory and practice*. Cambridge University Press, 2009.

[6] Q. Hu, G. Wang, and P. S. Yu. Assessing the longevity of online videos: A new insight of a video's quality. In *DSAA '14*, pages 1–10, 2014.

[7] Q. Hu, G. Wang, and P. S. Yu. Deriving latent social impulses to determine longevous videos. In *WWW' 14*, 2014.

[8] Q. Hu, S. Xie, S. Lin, W. Fan, and P. S. Yu. Frameworks to encode user preferences for inferring topic-sensitive information networks. In *SDM '14*, 2014.

[9] J. Karat. *Designing personalized user experiences in eCommerce*. Springer Science & Business Media, 2004.

[10] X. Kong, P. S. Yu, Y. Ding, and D. J. Wild. Meta path-based collective classification in heterogeneous information networks. In *CIKM '12*, 2012.

[11] P. McCullagh and J. A. Nelder. *Generalized linear models*. CRC press, 1989.

[12] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.

[13] B. D. Ripley. *Modern applied statistics with S*. Springer, 2002.

[14] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, 1999.

[15] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM '11*, 2011.

[16] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB '11*, 2011.

[17] G. Wang, Q. Hu, and P. S. Yu. Influence and similarity on heterogeneous networks. In *CIKM '12*, 2012.

[18] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. In *ICDM '11*, 2011.

[19] B. L. Welch. The generalization of 'student's' problem when several different population variances are involved. *Biometrika*, 1947.

[20] S. Xie, Q. Hu, J. Zhang, J. Gao, W. Fan, and P. S. Yu. Robust crowd bias correction via dual knowledge transfer from multiple overlapping sources. In *2015 IEEE International Conference on Big Data*, 2015.

[21] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM '14*, 2014.

[22] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han. Recommendation in heterogeneous information networks with implicit user feedback. In *RecSys '13*, 2013.

[23] J. Zhang, P. S. Yu, and Z.-H. Zhou. Meta-path based multi-network collective link prediction. In *KDD '14*, 2014.

[24] T. Zhu, P. Harrington, J. Li, and L. Tang. Bundle recommendation in ecommerce. In *SIGIR ' 14*, 2014.

[25] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.

[26] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML '03*, 2003.