Multi-Source Deep Learning for Information Trustworthiness Estimation

Liang Ge, Jing Gao, Xiaoyi Li and Aidong Zhang
Dept. of Computer Science and Engineering
The State University of New York at Buffalo
Buffalo, NY, USA
{liangge, jing, xiaoyili, azhang}@buffalo.edu

ABSTRACT

In recent years, information trustworthiness has become a serious issue when user-generated contents prevail in our information world. In this paper, we investigate the important problem of estimating information trustworthiness from the perspective of correlating and comparing multiple data sources. To a certain extent, the consistency degree is an indicator of information reliability–Information unanimously agreed by all the sources is more likely to be reliable. Based on this principle, we develop an effective computational approach to identify consistent information from multiple data sources. Particularly, we analyze vast amounts of information collected from multiple review platforms (multiple sources) in which people can rate and review the items they have purchased. The major challenge is that different platforms attract diverse sets of users, and thus information cannot be compared directly at the surface. However, latent reasons hidden in user ratings are mostly shared by multiple sources, and thus inconsistency about an item only appears when some source provides ratings deviating from the common latent reasons. Therefore, we propose a novel two-step procedure to calculate information consistency degrees for a set of items which are rated by multiple sets of users on different platforms. We first build a Multi-Source Deep Belief Network (MSDBN) to identify the common reasons hidden in multi-source rating data, and then calculate a consistency score for each item by comparing individual sources with the reconstructed data derived from the latent reasons. We conduct experiments on real user ratings collected from Orbitz, Priceline and TripAdvisor on all the hotels in Las Vegas and New York City. Experimental results demonstrate that the proposed approach successfully finds the hotels that receive inconsistent, and possibly unreliable, ratings.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Algorithms

KDD' 13, August 11–14, 2013, Chicago, Illinois, USA.

Keywords

Deep Learning, Multiple-Source, Information Trustworthiness

1 INTRODUCTION

With the booming of Internet applications and mobile devices, it is now much easier for people to access, create and publish contents than ever before, which leads to great information exposure for almost everyone in the world. Since everyone is able to generate contents online, as the side effect of freedom of speech, information trustworthiness has become a serious problem for many applications including Ebay [16], Twitter [10] and Amazon [7, 12]. One of the fundamental difficulties in analyzing user-generated data is that information is massive, yet can be noisy, incorrect and misleading. It is hard to infer reliable knowledge from massive data with low data quality, and there exists no universal oracle that can tell us which information source is reliable and which piece of information is trustworthy. Therefore, we seek to infer information trustworthiness from the data by cross checking multiple data sources. Trustworthiness of contents cannot be inferred purely based on its information alone, but rather by exploring consensus and commonalities in multiple sources simultaneously. A piece of information is more likely to be reliable if it is supported by many independent sources while high inconsistency across sources may suggest potentially unreliable information. In this paper, we propose to infer the consistency degree of information across multiple sources as an effective indicator for information trustworthiness.

Applications. Information trustworthiness is an important issue for numerous applications, but in this paper, we particularly focus on online recommendation systems, which typically involve overwhelming amounts of user-generated data. Nowadays, there are many places where people can leave their opinions for their experiences with products or services, in the form of ratings and reviews. Those ratings and reviews exert great influence on people before they make their decisions about their potential purchases. In fact, more and more people rely on such websites (e.g., Yelp) to find an apartment, book a hotel and reserve a table in a restaurant, etc. However, people are bothered by the existence of unreliable and misleading information especially when there exist fake reviews or ratings posted by spammers. Although some endeavors have been made to detect and prevent spam reviews and ratings [7, 9, 12, 13], people are still struggling with untruthful information.

In reality, people will check multiple websites for unbiased opinions before they make decisions because they are fully

aware of the biases and noises embedded in a single source. Intuitively, consensus opinions across multiple sources about an item should be valued more, but high discrepancy may indicate that the information is suspicious. However, the huge volume of data makes the task of manually checking multiple review websites time-consuming and sometimes impossible. Motivated by this observation, we propose to develop an effective and efficient approach to compare multiple sources of information about the same item and calculate a consistency score for each item to assist users in decision making. In this paper, we focus on *identifying consistent information from user rating matrices collected from multiple platforms.* The framework can be extended in the future to incorporate other information, such as user profiles and review comments.

Challenges. One major challenge in consistency degree computation is that different platforms attract different sets of users. It is impossible to simply compare ratings at the user level because users may leave ratings only at one place. Even if there are overlapping users, we are unable to align them due to the lack of users' information. On the other hand, comparing rating statistics or summary obtained by aggregating ratings in individual sources can also be problematic because users have quite diverse tastes and preferences. As users with different backgrounds all contribute to the ratings and reviews, the difference between the rating summaries of multiple sources could well reflect user preference diversity rather than information inconsistency. To accurately estimate consistency degrees, we must identify subtle commonalities shared among sources embedded in multiple rating matrices.

Observations. In regard to the challenges discussed above, we have the following observations that shed some lights on the problem.

- Many latent reasons contribute to the ratings that users gave. For example, many possible ways can be enumerated to explain the ratings of hotels, which include the purpose of the trip, type of the users (e.g. family travelers or solo travelers), services of the hotel, price, view and location of the hotel. Moreover, we notice that those latent reasons can be grouped into several categories, such as reasons about the traveler(s), reasons about the hotel and reasons about the location. Clearly, there exists a hierarchical latent structure underlying the ratings data that corresponds to various reasons leading to users' decisions.

- Looking at multiple sources simultaneously, we can find that users' latent rating behavior is consistent even though users are different across platforms. At the surface, ratings collected from different sources can have different rating scales and have diverse distributions; each source may have its own bias; and spammers may contaminate some of the entries. Despite all these facts, the latent reasons that account for majority of the users' ratings are consistent across sources. Based on this consensus principle, we are able to compute a consistency degree for each item indicating the chance this item receives consistent ratings across sources.

Summary. These observations motivate us to propose a novel two-step procedure to estimate information consistency across multiple sources given multiple rating matrices collected from different platforms. The key idea is to connect multiple sources by learning their shared latent hierarchy of rating reasons and then compare multiple sources at the latent reason layer to obtain the consistency scores. First, we develop a Multi-Source Deep Belief Network (**MSDBN**) to learn a joint model that represents the common hidden reasons underlying the observed ratings across sources. Since the latent space for each source forms a hierarchical structure, a deep network can be used to extract latent structure from each source. We connect multiple sources by linking the deep network structures through a joint consensus layer which represents the common hidden reasons embedded in multiple data sources. At the second step, we reconstruct latent representations for each source from the joint representation of multiple sources. The reconstructed data simulates users' rating behavior in each source if the source follows the common latent reasons. By comparing reconstructed data based on common reasons and each source's own representation, we can successfully derive information consistency for each item, which can be further used to suggest the trustworthiness of an item's information. Note that our work differs from existing deep learning approaches in that we develop an approach to compute information consistency across multiple sources while existing work focuses on single data source and targets at different problems (classification, clustering, etc.). To demonstrate the effectiveness of the proposed approach, we crawled real ratings of hotels in two big cities from three travel websites: Orbitz, Priceline and TripAdvisor. Experimental results on this multi-source data set show that the proposed method provides veracious estimation of information consistency to help users identify trustable information from massive, conflicting and biased data. We also design experiments on synthetic data to perform a thorough quantitative analysis illustrating the strong abilities of the proposed approach in distinguishing consistent from inconsistent information.

To sum up, the contributions of this paper are:

- We investigate the problem of information trustworthiness from the novel perspective of exploring multiple sources' information consistency degree. The proposed method simulates the natural intuition that more consistent information stated by many sources is more likely to be trustworthy.

- We propose an unsupervised Multi-Source Deep Belief Network to learn a joint model across sources to capture the consensus rating behavior in multi-source rating data. Based on the joint structure, we develop an effective approach to evaluate information consistency by comparing original and reconstructed data.

- Experimental results on both real and synthetic data sets show that the proposed approach is able to discover common rating behavior shared by multiple sources, derive each item's information consistency, and thus output meaningful alerts for inconsistent and unreliable information.

2 METHODOLOGY

In this section, we present the details of the proposed method. We first present the description of the problem formulation in Section 2.1. Section 2.2 discusses how to represent the latent reasons for a single source. The two-step procedure of information trustworthiness estimation is presented in Sections 2.3 and 2.4. The first step is to train a Multi-Source Deep Belief Network (MSDBN) so that the common latent reasons across sources are captured. The consistency score is

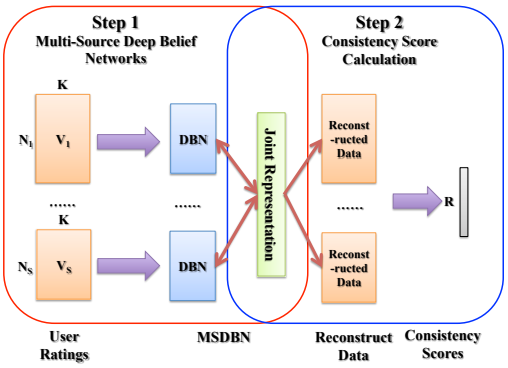


Figure 1: Flow of the Proposed Method

calculated accordingly in the second step. We discuss some practical issues of training MSDBN in Section 2.5.

2.1 Problem Formulation

Suppose we are interested in K items (each item could be a hotel, a book, a restaurant or any entity of interests). There are S sources that we can obtain ratings about the K items. The s-th source is characterized by a rating matrix V_s, which denotes ratings of K items from N_s users. Note that the users are differen across sources. The goal is to derive a consistency score vector R, where each entry r_k denotes the consistency score of the k-th item. The general intuition is that an item will receive high score if its ratings are consistent across multiple sources. The consistency score denotes our recognition that information unanimously agreed by all the sources is most likely to be reliable. Due to noise, sparsity and alignment issues, it is impossible to determine the consistency of items by directly comparing their ratings. However, latent reasons hidden in user ratings are mostly shared by multiple sources and inconsistency about an item can only be revealed when some source provides ratings deviating from the common latent reasons. Therefore, we propose a Multi-Source Deep Belief Network to extract common reasons underlying the observed rating matrices V_1,...,V_s. The trained MSDBN is used to reconstruct data for each source and the reconstructed data simulate the ratings following the common latent reasons. The consistency score for each item can thus be obtained by calculating the similarity between the original data and the reconstructed data across multiple sources. The flow of the proposed method is shown in Figure 1. Table 1 lists the notation used throughout this paper.

Table 1: Notation

Symbol	Definition
K	number of items
S	number of sources
V_s	rating matrix for s-th source
N_s	number of users in s-th source
U	weight matrix for source 1 in Sec. 2.3
V	weight matrix for source 2 in Sec. 2.3
b	bias for source 1 in Sec. 2.3
c	bias for source 2 in Sec. 2.3
a	bias for the top layer in Sec. 2.3
ϵ	learning rate for MSDBN
$h_s^{(n)}$	the n-th layer for s-th source
v_s	visible units for s-th source
d_{sk}	d_{sk} denotes the distance between original data and reconstructed data of k-th item in s-source

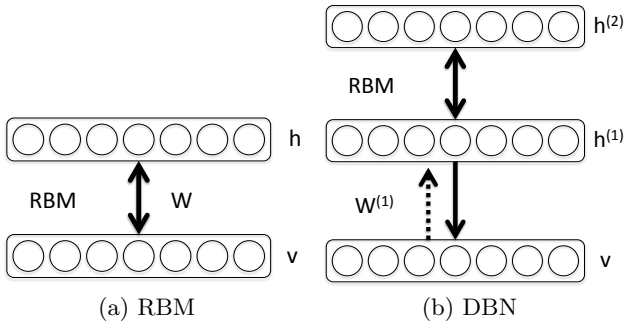


Figure 2: A Restricted Boltzmann Machine and A 2-Layer Deep Belief Network for Each Source

2.2 Single Source Representation

As we discussed in Section 1, there are many possible latent reasons to explain users' rating behavior. Therefore, given a rating matrix V_s for s-th source, how to represent the latent reasons underlying V_s is our first problem.

A common approach to represent the latent reasons is using clustering techniques [2,3,21] where V_s is modeled by the product of matrix $P_{N_s \times C}$ and $Q_{C \times K}$. Such model performs clustering on the input space where P is the clustering indication matrix and Q is the cluster level feature (C clusters are obtained). Users in the same cluster in P share similarity in terms of their rating behavior, and the aggregated rating of the cluster is captured by Q. One limitation of the clustering techniques is that it usually forms a coarse representation of the latent reasons. Users in the same cluster are similar because of combinations of several latent reasons. For example, the reason that users belong to the same cluster may be because they are all family travelers who like bargain hotels and free wifi. Increasing the number of clusters doesn't usually get a finer representation, because it will usually create a lot of trivial clusters with very few users in them.

To have a finer representation of latent reasons for a single source, we propose to use Restricted Boltzmann Machine (RBM). RBM is an undirected graphical model with visible units v and stochastic binary hidden units h. The visible units v denote the observed data (in our case, V_s) and the hidden units h denote the latent reasons that generate the observed data. There are symmetrically weighted connections W between each visible unit and each hidden unit. There is no connections between visible units and between hidden units. Therefore, RBM forms a bipartite graph between visible and hidden units as illustrated in Figure 2 (a).

The probability distribution of RBM is as follows:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)},$$

where the partition function Z is given by summing over all possible pairs of visible and hidden vectors: $Z = \sum_{v,h} e^{-E(v,h)}$. The energy function of visible and hidden units is

$$E(v,h) = -\sum_{i \in visible} b_i v_i - \sum_{j \in hidden} a_j h_j - \sum_{i,j} v_i h_j w_{ij}, \quad (1)$$

where a_i, b_j are biases for hidden and visible units and w_{ij} is the weight between them. The purpose of RBM is to find a configuration of (v,h) so that the energy function achieves its lowest level.

Since RBM takes the shape of a bipartite graph, with no direct connections between hidden units and between visible units, the hidden units are mutually independent given

Algorithm 1 MSDBN Training Algorithm For Two Sources

Input: Input for each source M_1 and M_2, learning rate ϵ
Output: Weight matrices of two sources–U and V, biases of two sources–b and c, top layer bias a

Randomly initialize U, V, a, b and c
repeat
  repeat
    pick up a sample x_1 from M_1 and sample x_2 from M_2
    for all hidden units i **do**
      compute $P(h_{1i} = 1|x_1, x_2)$ using Eq. 8
      sample h_{1i} from $P(h_{1i}|x_1, x_2)$
    end for
    for all visible units j in source 1 **do**
      compute $P(x_{2j}^{(1)} = 1|h_1)$ using Eq. 6
      sample $x_{2j}^{(1)}$ from $P(x_{2j}^{(1)}|h_1)$
    end for
    for all visible units j in source 2 **do**
      compute $P(x_{2j}^{(2)} = 1|h_1)$ using Eq. 7
      sample $x_{2j}^{(2)}$ from $P(x_{2j}^{(2)}|h_1)$
    end for
    for all hidden units i **do**
      compute $P(h_{2i} = 1|x_2^{(1)}, x_2^{(2)})$ using Eq. 8
    end for
  until for all samples in M_1 and M_2
  $U \leftarrow U + \epsilon(h_1 x_1' - P(h_2 = 1|x_2^{(1)})x_2^{(1)'})$
  $V \leftarrow V + \epsilon(h_1 x_2' - P(h_2 = 1|x_2^{(2)})x_2^{(2)'})$
  $a \leftarrow a + \epsilon(h_1 - P(h_2 = 1|x_2^{(1)}, x_2^{(2)}))$
  $b \leftarrow b + \epsilon(x_1^{(1)} - x_2^{(1)})$
  $c \leftarrow c + \epsilon(x_1^{(2)} - x_2^{(2)})$
until all parameters are converged
return U, V, a, b and c

the visible units and vice versa. The individual activation probabilities of a hidden and a visible unit are given by

$$P(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}), \tag{2}$$

$$P(v_i = 1|h) = \sigma(a_i + \sum_j h_j w_{ij}), \tag{3}$$

where σ denotes the logistic sigmoid function. Considering Eq. 2, suppose h_j is one latent reason for ratings in V_s (e.g., price), the activation probability $P(h_j = 1|v)$ shows how "important" this reason is given the observed data. Given many hidden units (e.g., 500) that represent the latent reasons, the learning of RBM could be understood as tuning up the *importance* of all latent reasons given the observed data set so that the hidden units could get close to the true latent reasons as much as possible. For RBM, exact maximum likelihood learning is intractable. In practice, efficient learning is performed using Contrastive Divergence (CD) [4].

The advantage of RBM is that it discovers a richer representation of the input data than clustering techniques. Each hidden unit in RBM creates a 2-region partition of the input space and n hidden units can represent up to 2^n different regions in input space. This indicates that given L latent reasons in input space, clustering techniques will take L parameters (e.g., number of clusters) to capture that many reasons, RBM only need $log_2(L)$ hidden units. Since there are many possible latent reasons underlying V_s for s-th source, we choose RBM instead of clustering techniques to represent each single source.

RBM can represent many possible underlying reasons of each source, such as reasons about hotels and reasons about location as we discussed in Section 1. Within each type of reasons, there exist many reasons about the detailed aspects. For example, one may choose a hotel due to its room

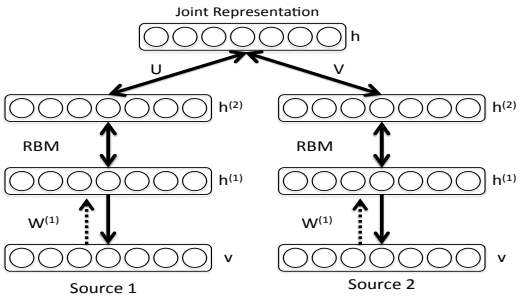


Figure 3: Multi-Source Deep Belief Network

service, staff quality, or internet services. We can go even further down the hierarchy for more detailed reasons that contribute to users' ratings. Therefore, these reasons form a hierarchical structure, and we can thus add more layers into RBM to form a Deep Belief Network (DBN) to represent such complicated latent reasons. A DBN with l layers models the joint distribution between observed variables v and l hidden layers $h^{(k)}$, $k = 1, ..., l$ as follows:

$$p(v, h^{(1)}, ..., h^{(l)}) = P(v|h^{(1)})...P(h^{(l-2)}|h^{(l-1)})p(h^{(l-1)}, h^{(l)}), \tag{4}$$

Denoting $b^{(k)}$ the bias vector of layer k and $W^{(k)}$ the weight matrix between layer k and $k+1$, we have

$$P(h^{(k)}|h^{(k+1)}) = \Pi_i P(h_i^{(k)}|h^{(k+1)}),$$

$$P(h_i^{(k)} = 1|h^{(k+1)}) = \sigma(b_i^{(k)} + \sum_j W_{ij}^{(k)} h_j^{(k+1)}).$$

A two-layer DBN is shown in Figure 2 (b). The training of DBN follows a greedy layer-wise CD strategy [4].

2.3 Multi-Source Deep Belief Network

Although we extract latent reasons of each source using DBN, we can't obtain the consensus reasons by directly comparing them because 1) The latent reasons of each source may contain their source-specific bias, making them hardly comparable across sources. 2) For the common latent reasons each source contain, they are not properly aligned. Therefore, to find the common latent reasons across sources, we propose the Multi-Source Deep Belief Network (MSDBN).

We illustrate the construction of a MSDBN using two sources as a running sample. Note that it can be easily extended to accept inputs from multiple sources. Consider modeling each source using a two-layer DBN. The energy function $P(v, h^{(1)}, h^{(2)})$ is given by Eq. 4. To form a MSDBN, we combine the two DBNs by adding an additional layer of binary hidden units on top of them. The resulting graphical model is shown in Figure 3. The joint distribution over the multiple sources can be written as:

$$P(v_1, v_2, h) = P(h_1^{(2)}, h_2^{(2)}, h)P(v_1, h_1^{(1)}, h_1^{(2)})P(v_2, h_2^{(1)}, h_2^{(2)}),$$

where $P(h_1^{(2)}, h_2^{(2)}, h)$ can be written as follows:

$$P(h_1^{(2)}, h_2^{(2)}, h) \propto \exp(\sum_i a_i h_{1i}^{(2)} + \sum_j b_j h_{2j}^{(2)} + \sum_k c_k h_k + \sum_{i,k} h_{1i}^{(2)} U_{ik} h_k + \sum_{jk} h_{2j}^{(2)} V_{jk} h_k), \tag{5}$$

where U and V are the weight matrix connecting the top hidden layer of DBN for each source. a, b and c are the corresponding bias vectors. Given the MSDBN, the conditional

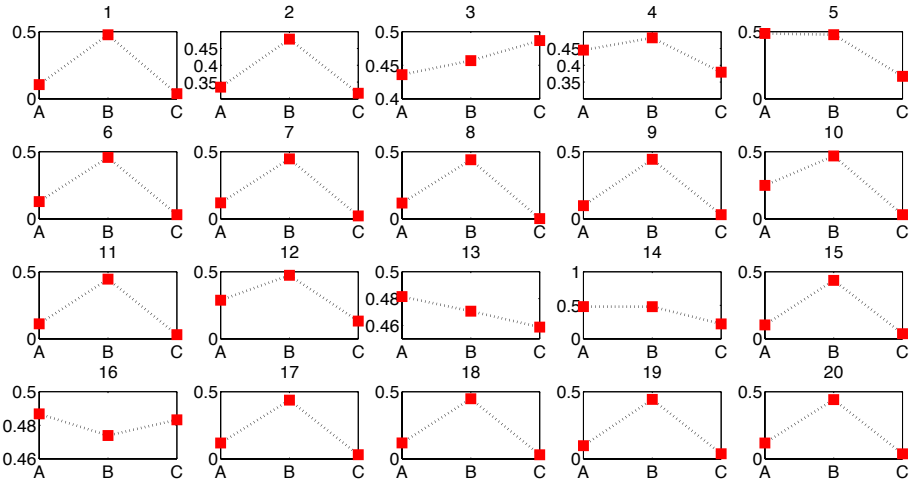


Figure 4: The distance between original data and reconstructed data across three sources for 20 hotels in Las Vegas. X axis denotes the three sources with A: Orbitz, B: Priceline and C: TripAdvisor. Y axis denotes the distance between original data and reconstructed data. The majority of hotels exhibit a similar shape.

distribution is derived as follows:

$$P(h_1^{(2)}|h) = \sigma(a + \sum_i h_i U_i), \tag{6}$$

$$P(h_2^{(2)}|h) = \sigma(b + \sum_i h_i V_i), \tag{7}$$

$$P(h|h_1^{(2)}, h_2^{(2)}) = \sigma(c + \sum_i U_i h_{1i}^{(2)} + \sum_j V_j h_{2j}^{(2)}), \tag{8}$$

where σ is logistic sigmoid function. The learning of MSDBN is to tune up the top hidden layer h (Eq. 8) so that it can better generate the input sources (Eq. 6 and Eq. 7). In this way, MSDBN tries to find the shared latent reasons that underly multiple sources.

The training of MSDBN using two sources is shown in Algorithm 1. In Algorithm 1, lines 5 to 19 prepares ingredients for CD [4]. Specifically, lines 5 to 8 compute the activation probability of hidden units of MSDBN based on two input sources; lines 9 to 16 reconstruct source 1 and 2 based on hidden units; and lines 17 to 19 calculate the activation probability of hidden units based on two reconstructed sources. Lines 21 to 25 update parameters U, V, a, b, c accordingly. The algorithm stops when all parameters are converged.

2.4 Consistency Score Calculation

At the first step, we obtain the consensus latent reasons underlying multiple sources. Consequently, at the second step, based on the learned consensus hidden reasons, we estimate the information trustworthiness by calculating a consistency score for each item. The higher the score, the more consistently the item behaves across multiple sources, the more likely that the information about the item is reliable.

Once the MSDBN is trained, we have the top layer of hidden units that represents the consensus reasons. Next, we reconstruct each source using Eq. 6 and Eq. 7. The reconstructed data are sampled from $P(h_1^{(2)}|h)$ and can be viewed as the data generated from the consensus reasons in each source. Therefore, for a given source s and item i, we have its original data and its reconstructed data. We calculate their distance using Root Mean Square Error (RMSE) to form a matrix D of size $S \times K$ where d_{sk} represents the distance between original data and reconstructed data of item k on source s.

Ideally, if an item behaves consistently across sources, the distance between its reconstructed data and its original data should be small since both of them are driven by the same consensus reasons. However, the reconstruction of each source by MSDBN is not perfect, i.e., there is reconstruction error for each source. The existence of reconstruction error for each source severely prohibits us to simply measure consistency scores using distance aggregated upon multiple sources. An item that has a large overall distance between original and reconstructed data doesn't necessarily mean that the information it receives is inconsistent.

To tackle this challenge, we will examine the distance matrix D to find the consensus patterns as a basis for consistency score computation because majority of the items receive consistent user ratings across sources. Figure 4 plots the distance between the original data and reconstructed data for 20 hotels in Las Vegas from three sources: Orbitz, Priceline and TripAdvisor. As we can see, for the majority of items, the distance between original data and reconstructed data follows a similar shape. This shape has close relations with reconstruction error of each source, which can be seen as the reflection of the bias of each source in terms of the consensus latent reasons. For items that have a different shape than that of the majority items (e.g., items 3 and 13), we believe it is highly possible that some inconsistency resides among them.

Since the majority of items are consistent, we thus derive the **Consistency Score** r_k of each item k by following:

$$p = median(D, 2), \tag{9}$$

$$r_k = Similarity(D(:, k), p), \tag{10}$$

where Eq. 9 takes the median of each column in D, and thus p stands for the distance between original data and reconstructed data for the majority of consistent items. The consistency score r_k for each item k is consequently obtained by calculating the similarity between each row of D with p. In this work, we use Pearson Correlation as the similarity measure as it focuses on the shape similarity rather than the absolute distance. The lower the score, the further the item is away from consistent items, which indicates a higher possibility that the item receives inconsistent information.

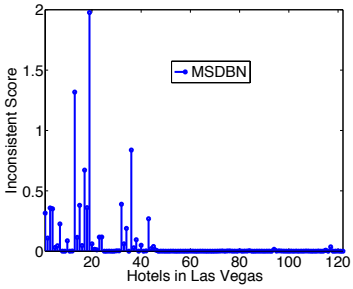


Figure 5: Inconsistent Score Distribution of Hotels in Las Vegas

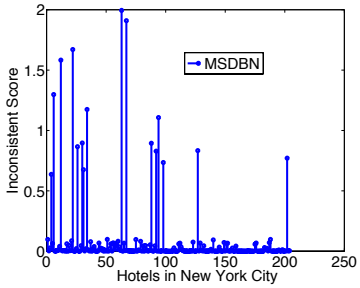


Figure 6: Inconsistent Score Distribution of Hotels in NYC

Since the major part of the proposed method lies in the construction of Multi-Source Deep Belief Network, we name our method **MSDBN**.

2.5 Practical Issues

In MSDBN, DBN for each source has binary visible units yet the rating matrix is count data whose range is from 0 to N. As suggested in [18], one simple and effective way is to make N copies of binary visible units and give them all the same weights and biases. Using this weight-sharing to synthesize count data out of binary units will keep the mathematics underlying binary-binary DBN unchanged.

MSDBN accepts inputs from DBN of each source. The DBN of each source could contain several layers of hidden units. The number of layers and the number of hidden units are strongly related to the representational power of MSDBN. Increasing the number of layers can also reduce the number of parameters used in the model. However, there is a trade-off between the performance and time of training in terms of the hidden number of units and the number of layers. In this work, we maintain the number of layers of DBN for each source to be 2 and the number of hidden units on the top layer of MSDBN to be 500.

There are some issues involved in training MSDBN in Algorithm 1 including choosing the learning rate ϵ, initialization of the weights and biases and stopping criteria. A detailed guide on these issues can be found in [5].

Table 2: Hotel Rating Data Sets: Las Vegas

Features	Orbitz	Priceline	TripAdvisor
#. of Users	34,735	2,530	100,037
Avg. Hotel Rate #.	1407.4	423.5	3506.7
Avg. User Rate #.	5.3	21.9	4.5
Avg. Hotel Rate	3.6	3.7	3.8
Rate Variance	1.3	3.6	1.4

Table 3: Hotel Rating Data Sets: New York City

Features	Orbitz	Priceline	TripAdvisor
#. of Users	10,259	3,096	117,582
Avg. Hotel Rate #.	230.6	363.7	2172.1
Avg. User Rate #.	5.5	29.2	4.6
Avg. Hotel Rate	3.9	3.9	4.0
Rate Variance	1.0	2.9	1.3

3 EXPERIMENTS ON REAL DATA SETS

In this section, we apply the proposed method on the real hotel rating data sets and show how the proposed approach issues meaningful alerts on unreliable information.

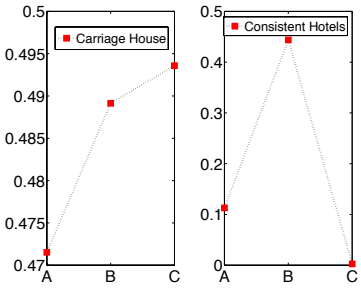


Figure 7: The distance between original data and reconstructed data for Carriage House and consistent hotels in Las Vegas

3.1 Data Sets

The two data sets are the ratings of hotels crawled from three popular travel websites in United States: Orbitz, Priceline and TripAdvisor. Two popular cities, Las Vegas and New York City are chosen in our experiments because there are plenty of hotels being reviewed. The three websites have different numbers of hotels in the two cities and we crawl all the ratings of the common hotels among the three websites. The data sets are crawled between March 7 and March 9, 2012. The rating scale is between 1 and 5. Tables 2 and 3 show the characteristics of the two real data sets.

3.2 Results and Evaluation

The output of MSDBN is a consistency score vector R. Since it is more interesting to present some inconsistent information, we compute the **Inconsistency Score** for each item as $I(k) = \alpha - r_k$ where α is the maximal value in R. We apply the MSDBN on the above data sets and calculate the inconsistency score for each hotel. Figures 5 and 6 show the inconsistency score distribution of hotels in Las Vegas and New York City.

There are several observations that can be drawn from the figures. First, most hotels in Las Vegas and New York city receive low inconsistency scores, indicating that the information about most hotels in three websites are consistent. If ratings about a hotel are consistent across multiple sources, we can usually claim that the information about this hotel are reliable. Second, the inconsistency score for some hotels are significantly higher, indicating that there exist large inconsistency about the information of these hotels across sources. This usually requires further investigations about these hotels to determine if their information are reliable or not. Next, we present two case studies to show that there indeed exist unreliable information in their ratings. Although we don't have ground truth for this task, we find substantial evidence to support the findings of the proposed method.

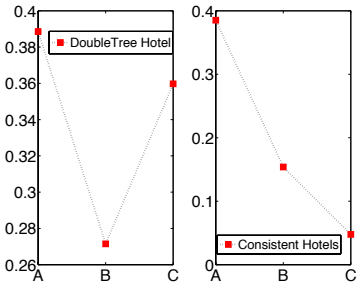


Figure 8: The distance between original data and reconstructed data for DoubletTree Hotel and consistent hotels in NYC

Case Study I: In the first cast study, we pick the hotel that has the highest inconsistency score in Las Vegas: the Carriage House hotel. Figure 7 shows the distance between original data and reconstructed data for Carriage House and consistent hotels in Las Vegas from three sources. From the figure, it seems that the information about this hotel on the third source (TripAdvisor) rings a bell.

From TripAdvisor's perspective, Carriage House is a real nice hotel. It ranked 12-th of all 282 hotels in Las Vegas. More than 80% of users gave rating more than 4. However, other sources tell a different story. In Yelp, 50% of guests gave ratings less than 3. In Booking, 24.4% people gave ratings less than 4. Other than ratings, we summarize some of the reviews about the Carriage House from various sources in Table 4. As shown in the table, the Carriage Hotel shows many unattractive features and those features (e.g., loud, rude and dirty) are unfortunately quite consistent across multiple sources. This case study shows that the proposed method successfully detects the large inconsistency of information between TripAdvisor and other sources and can warn potential customers on the information trustworthiness of the ratings.

Table 4: Summary of Reviews about The Carriage House

Websites	Review Summary
Yelp	**super loud**, **unfriendly**, constant noise, **dirty**
Priceline	**rude** front desk, carpet **dirty**, AC **super loud**
Booking	**noisy**, **rude** staff, AC noisy, lazy service, **dirty**

Cast Study II: MSDBN discovers that the Double Tree Hilton Hotel near Time Square in New York city has the highest inconsistency score. Figure 8 shows the distance between original data and reconstructed data for Double Tree Hotel and consistent hotels in New York city. As we can see, in this case, information on TripAdvisor causes the high inconsistency again. Priceline lists this hotel as 11-th of the 357 hotels in New York city. In Booking, the overall rate is 8.5 out of 10 and it is considered very good among hotels in New York city. However, TripAdvisor ranks this hotel as as 256-th of 432 hotels in New York city and 40% of the rating are less than 3. Such huge contrasts between TripAdvisor and other sources remind us to be cautious about the information about this hotel.

The two case studies both point out that TripAdvisor is more likely to receive unreliable information from users. This makes sense in that TripAdvisor are totally open to anyone who is able to register, which is much easier to attract spam information. Note that the above results do not indicate that TripAdvisor is always less reliable compared with other sources, rather, it provides less reliable information on the hotels in the case studies.

4 EXPERIMENTS ON SYNTHETIC DATA

In Section 3, we present case studies on the real hotel rating data sets to show that the proposed method is effective in estimating the information trustworthiness in recommendation systems. In this part, we conduct experiments on the synthetic data sets to perform quantitative analysis on the proposed method.

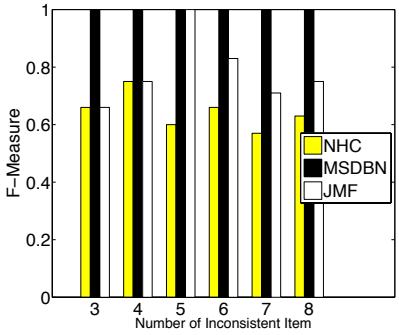


Figure 9: Performance Comparison

4.1 Data Generation and Evaluation Metric

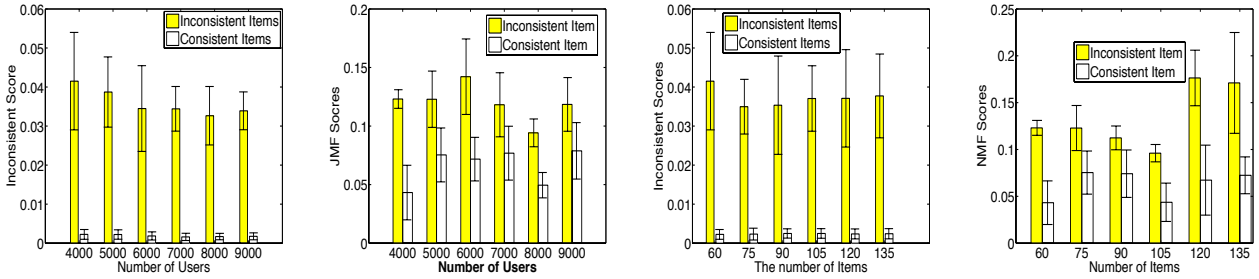
The synthetic data are generated based on the observation that the latent reasons across multiple sources are consistent. We generate three sources and mandate that they share a mixture of three rating distributions. For a given item, its ratings in each source are drawn from one of the distributions. t items are randomly chosen to be the items receiving inconsistent information, and their ratings are randomly shuffled. All the generated ratings are padded with random noises to simulate the fact that users have diverse preference.

The evaluation metric of this experiment is F-measure, defined as $2(precision \times recall)/(precision+recall)$. $precision$ defines the percentage of correctly identified items that receive inconsistent information among all the top t items returned by the method, and $recall$ is the percentage of correctly identified items that receive inconsistent information among the true top t items generated by the data generator.

4.2 Performance Comparison

To illustrate the power of the proposed model, we first introduce some baseline methods. As discussed in Section 1, the ratings from individual users can't be compared directly, yet we can consider the method which compares ratings statistics. The first baseline method is Normalized Histogram Comparison (NHC) as follows. For a source s and item k, we compute the percentage of users who give rating to item k from 1 to N (the maximum rating), and thus we have a rating summary vector of length N for each item in each source. Then we compute the inconsistency score as the minimum of the mutual distance among those summary vectors and pick up the top t items as items that receive the most inconsistent information. Note that mutual distance could be any distance measurement that fits the application. In this work, we use the commonly used Euclidean distance as the distance measure.

The second baseline method is from [3], which targets at the similar problem using Joint Matrix Factorization (JMF). JMF finds the consistent groups across multiple sources us-



(a) MSDBN w.r.t # Users (b) JMF w.r.t # Users (c) MSDBN w.r.t # Items (d) JMF w.r.t # Items

Figure 10: Performance comparison between MSDBN and JMF wrt # users and # items

ing joint matrix factorization and retrieves the inconsistent items by comparing items at group level.

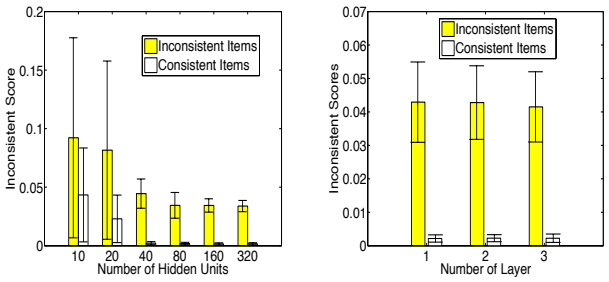
In this experiment, we compute the inconsistency score the same way as in Section 3. Figure 9 shows the performance comparison between NHC, JMF and MSDBN. It is obvious that NHC fails to detect some items receiving inconsistent information. It only compares high-level statistics, thus vulnerable to noises and has limited discriminative power. JMF performs better mainly because it considers group-level information. However, comparing with the proposed MSDBN, some inconsistent items are still undetected by JMF. This gap in performance between JMF and MSDBN is mainly due to the difference in representational power between clustering techniques and MSDBN, i.e., MSDBN explores a much finer representation of common latent reasons than clustering techniques.

4.3 Various Learning Scenarios

In this part, we show how MSDBN performs in various learning scenarios by tuning four variables: 1) the number of users, 2) the number of items, 3) the number of hidden units on the top layer of MSDBN, and 4) the number of layers in MSDBN. The first and second variables are used in synthetic data generation while the latter two are parameters used in MSDBN algorithm. We set the default settings of the four variables as follows. There are 4000 users and 60 items in each of the 3 sources. The rating scale is 1 to 5. Also we set number of inconsistent items $t = 7$, the number of hidden units $h = 500$ and the number of layers $l = 1$. Note that l is the layer of DBN for each source. When $l = 1$, DBN is reduced to RBM. While we vary one variable or parameter, we maintain others the same as in the default setting.

We present the experimental results in the following way. We partition the items into two sets: items that receive consistent and inconsistent information (referred to as **consistent** and **inconsistent sets**), which we know from data generation. Then in each figure, the bar represents the average inconsistency score for each set, and the vertical line denotes the variance of the scores in each set. *The method performs well if the difference of scores in the two sets is big*, which indicates its capability of separating items that have consistent and inconsistent ratings in multiple sources.

Number of Users and Items. When we vary the number of users, Figures 10 (a) and (b) show the performance comparison between MSDBN and JMF. As we can see, MSDBN exhibits great power of distinguishing inconsistent and consistent items in that the score difference in the two sets is rather significant. On the other hand, the performance of JMF is clearly inferior to that of MSDBN in that there exist overlapping of scores between inconsistent items and consistent items in some experiments of JMF. In addition,



(a) MSDBN w.r.t # Hidden Units (b) MSDBN w.r.t # Layers

Figure 11: Performance of MSDBN wrt # hidden units and # layers

in Figure 10 (a) we notice that the variance is reduced when the number of users increases, which indicates that the MSDBN produces better results. The improvement is due to the fact that when the number of users increases, the consensus latent reasons are becoming more and more dominant, rendering it easier to be captured by the MSDBN.

As for the number of items, Figures 10 (c) and (d) show the performance of MSDBN and JMF. Similar to the experiments with the number of users, inconsistent items are easily detected by the proposed MSDBN and the difference between consistent item set and inconsistent item set is significantly larger than that of JMF, indicating MSDBN outperforms JMF.

Number of Hidden Units and Layers. The number of hidden units on the top layer of MSDBN is closely related to the representational power of MSDBN. Figure 11 (a) shows the performance of the MSDBN in terms of the number of hidden units. As we can see from the Figure, when the number of hidden units is small, e.g., 10, the performance of MSDBN is poor in that the scores for consistent and inconsistent items are overlapping. When we increase the number of hidden unit to 40, MSDBN is able to distinguish inconsistent items from consistent items. When we further increase the number of hidden units, the performance is stable. This indicates that for the synthetic data, having hidden units over 40 is sufficient to model the variants in the input space and thus find the inconsistent items. Note that real-life data is more complicated than the synthetic data, so a larger number of hidden units should be chosen. However, there is a trade-off between the performance increase and training time in terms of the number of hidden units in that large number of hidden units implies long training time. For the number of layers, Figure 11 (b) shows the performance of the MSDBN in terms of the layers. As we can see, the performance is rather stable, indicating for synthetic data, one-layer of MSDBN is enough. However, in the real rating

data, we notice the hierarchical structure in the latent space and a two-layer MSDBN is thus chosen.

5 RELATED WORK

The proposed model is built on Restricted Boltzmann Machines, which, in recent years, have attracted many attentions [6,14,18,19] (to name a few). The detailed introduction can be seen in [1]. Our work shares similarity with Multimodal Deep Learning [8, 15, 20], where deep belief networks are trained on the image and text inputs to learn a joint representation to perform generative and discriminative tasks. Different from these studies, in our work, the learnt joint representation denotes the consistent latent reasons that underly users' ratings from multiple sources. We thus utilize the common latent reasons to calculate consistency score for each item.

The work in [3] targeted the similar problem that estimates the local information trustworthiness and proposed a Joint Matrix Factorization (JMF) method to connect multiple sources. JMF useed clustering techniques to capture the variances of multiple sources which forms a coarse representation of the input space and thus has inferior discriminative power to the proposed method.

Information trustworthiness is a serious problem in various applications such as online auction website (e.g., Ebay) [16], social networks (e.g., Twitter) [10], and product reviews (e.g., Amazon) [7, 12]. Most of the work detect spammer or spam information based on single source of data. Moreover, some studies formulate the problem of information trustworthiness into a supervised task where labeled information are required for training. Our work focus on the information trustworthiness estimation from multiple sources and works in a pure unsupervised manner.

In the truth discovery field [11,17,22] , people work on the problem of detecting the truth about some questions or facts given multiple conflicting sources. In these studies, truth is considered as the fact that is told by many reliable sources and sources that often tell the truth is considered as reliable. Different from these truth discovery approaches, the goal of our work is to give a consistency score for each item across sources, indicating the trustworthiness of information about each item.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed to tackle the problem of information trustworthiness estimation by modeling the common latent reasons across multiple sources and computing a score for each item to quantify the degree of inconsistency in the information it receives. A novel two-step procedure was proposed to solve the problem: A Multi-Source Deep Belief Network (MSDBN) is first constructed to learn a joint representation that underlies ratings across multiple sources, and then each source is reconstructed based on the joint representation and an item's consistency score is computed based on the degree to which its actual ratings are aligned with reconstructed data. In real datasets collected from three popular travel planning websites on Las Vegas and New York city hotels, we showed that the proposed method discovered hotels that receive inconsistent and possible unreliable information. Quantitative analysis on synthetic data demonstrated the superior performance of the proposed method compared with other baseline methods in distinguishing inconsistent from consistent information.

Note that time factors play an important role in information trustworthiness analysis. In the hotel review example, hotels' conditions, users' preferences and spammers' strategies all change over time, so information quality of sources also changes. In the future, we plan to consider the effect of time and estimate the information trustworthiness as time evolves. Furthermore, we hope that our information trustworthiness analysis tool can help in flagging spam reviews. To achieve this goal, we should utilize fine-grained text analysis in the model and this is the other research direction we want to explore.

7 References

[1] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2009.

[2] S. Bickel and T. Scheffer. Multi-view clustering. In *International Conference on Data Mining*, 2004.

[3] L. Ge, J. Gao, X. Yu, W. Fan, and A. Zhang. Estimating local information trustworthiness via multi-source joint matrix factorization. In *International Conference on Data Mining*, 2012.

[4] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

[5] G. Hinton. A practical guide to training restricted boltzmann machines. Technical report, The Unversity of Toronto, 2010.

[6] G. Hinton, S. Osindero, and W. Teh. A fast learning algorithm for deep belief netts. *Neural Computation*, 2006.

[7] N. Jindal and B. Liu. Opinion spam and analysis. In *International Conference on Web Search and Data Mining*, 2008.

[8] Y. Kang and S. Choi. Restricted deep belief networks for multi-view learning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011.

[9] S. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *International World Wide Web Conference*, 2004.

[10] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: Social honeypots+machine learning. In *International Conference on Information Retrieval*, 2010.

[11] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Sirvastava. Truth finding on the deep web: Is the problem solved? *PVLDB*, 2013.

[12] E. Lim, V. Nguyen, N. Jindal, B. Liu, and H. Lauw. Detecting product review spammers using rating behaviors. In *International Conference on Information and Knowledge Management*, 2010.

[13] B. Mehta. Unsupervised shilling detection for collaborative filtering. In *Advancement of Artificial Intelligence*, 2007.

[14] A. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. In *Advances in Neural Information Processing Systems workshop*, 2009.

[15] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *International Conference on Machine Learning*, 2011.

[16] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos. Netprobe: A fast and scalable system for fraud detection in online auction networks. In *International World Wide Web Conference*, 2007.

[17] J. Pasternack and D. Roth. Making better informed trust decisions with generalized fact-finding. In *International Joint Conference on Aritifical Intelligence*, 2011.

[18] R. Salakhutdinov and G. Hinton. Replicated softmax: An undirected topic model. In *Advances in Neural Information Processing Systems*, 2009.

[19] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *International Conference on Machine Learning*, 2007.

[20] N. Sirvastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, 2012.

[21] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. In *Advancement of Artificial Intelligence*, 2009.

[22] X. Yin, J. Han, and P. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, 2008.