RANN: A Reliability-Aware Neural Network for Mining Unreliable Multidimensional Data

Abstract

This paper proposes a novel *Reliability-Aware Neural Network* (RANN) for mining unreliable multidimensional data, which are atypical of online user-generated content. In the pre-processing stage prior to the mining step, the *reliability* of each dimension of each data sample is first assessed. The proposed RANN subsequently mines the input data with the awareness of the unreliability among different dimensions of every data sample. To derive the output of the proposed model when it is applied onto a data sample, both the value of the input data and the reliability thereof are sequentially propagated between adjacent layers of the RANN. To learn a RANN from a set of training data, a modified backpropagation algorithm is custom designed by extending the classical backpropagation algorithm. To explore the capability of the proposed RANN method in mining unreliable data, we examined the performance of RANN in comparison with a series of peer methods in predicting values of target variables using two popular online user review datasets. Experimental results show that the new method consistently outperforms state-of-the-art algorithms in accomplishing the prediction tasks, which demonstrates its superiority in mining unreliable multidimensional data.

1 Introduction

User generated content is proliferating at a steady and increasing pace, leading to its abundant availability on the Internet. Such content often carries valuable insights for understanding consumer behaviors, overviewing population dynamics, and discovering emerging market trends. Unfortunately, the quality of such information can vary significantly, causing a major obstacle for systematically leveraging these rich and diverse resources for reliable knowledge discovery. Due to the significance of this problem, uncertain data mining and management has emerged as a topic of heated research interest recently [Aggarwal, 2010]. In particular, for supervised learning tasks, if unreliable and reliable data are not processed in a differentiated manner, either as labeled ground-truth records or as input for testing purpose, a trained model tends to suffer in both its prediction accuracy and reliability [Qin *et al.*, 2009b; 2009a]. To address this problem, this study proposes a novel *Reliability-Aware Neural Network* (RANN) for mining unreliable data, which are atypical of online user-generated content vastly and pervasively available today. Given a source of web data, the new method first assesses the *reliability* of each dimension of each input data sample. The gauged data reliability information is then carefully observed during the input data propagation and transformation process carried out by the RANN to derive the target outcome.

It is noted that the traditional practice to deal with unreliable data usually assumes that each data sample carries an overall reliability or quality score [Aggarwal and Yu, 2009] so that a weighting scheme, which is performed on *the level of individual data samples*, can discriminate the respective impact on an algorithm's decision making by a collection of data samples, each of which may bear a different level of reliability. For example, in probabilistic databases [Dalvi and Suciu, 2007], each data tuple can be associated with a probability number that indicates the confidence or reliability of the tuple. Such conventional practice treats all dimensions of a data sample as of a uniform quality. Despite the usefulness of this tuple-level reasoning about uncertainty for solving certain types of problems where the quality of information varies from tuples to tuples, the diversity data sources available on the Internet suggests a related but different problem in that each dimension of each data sample may possess its own reliability, which may be due to the way how the information is acquired. For example, information about a product can be aggregated from multiple online sources where each source carries its own data quality. In those scenarios, the quality of information varies not only across data points, from tuples to tuples, but also within each individual multidimensional data point, from dimensions to dimensions.

Recognizing the above need for uncertainty reasoning regarding online multidimensional data of diverse reliabilities, we proposed RANN in this study as a potential solution to address the new challenge, which is pervasively useful for processing information acquired from the Internet, such as mining user generated online content or information aggregated from multiple online sources where the reliability of information tends to vary from users to users and from sites to sites. Empowered by the comprehensive and in-depth awareness of reliability among each dimension of each multidimensional data sample, the proposed RANN is able to train itself through learning from labeled training data by differentiating the reliability of input data on the fine granularity of individual dimensions of individual data samples.

The rest of this paper is organized follows. We first review some most related work to this study in Sec. 2. Then we introduce the proposed RANN model and its custom designed training algorithm in Sec. 3. In Sec. 4, a series of experimental results is presented to explore and demonstrate the performance advantage of RANN in comparison with a set of peer methods for mining unreliable multidimensional data where the experimental data are adopted from two popular online user ratings datasets. Finally, we conclude this study in Sec. 5.

2 Related Work

A comprehensive survey regarding algorithms and applications on uncertain data mining can be found in [Aggarwal and Yu, 2009]. Topics attracting heated recent research interests in mining unreliable data include clustering [Lee *et al.*, 2007; Cormode and McGregor, 2008], frequent item mining [Chui *et al.*, 2007; Chui and Kao, 2008], and classification [Zhang, 2005; Qin *et al.*, 2009b; 2009a; Tsang *et al.*, 2011]. In this paper, we focus on supervised learning tasks involving unreliable data due to the wide applications of these tasks in reality. A main line of research is to extend traditional learning methods to incorporate the concept of uncertainty or unreliability in their algorithmic decision making. For example, researchers have extended rule-based classification algorithms [Qin *et al.*, 2009b], decision trees [Qin *et al.*, 2009a; Tsang *et al.*, 2011], and support vector machines [Zhang, 2005] to process data with uncertainty. Besides augmenting individual algorithms to cope with data uncertainty, associative classifiers combining extensions of multiple basic classifiers for uncertain data reasoning have also been proposed [Qin *et al.*, 2010; Minku and Yao, 2012]. Common to all methods overviewed above is that they assess the data reliability on the level of individual data samples and ignore the disparity of reliability among multiple dimensions of a data sample, which presents a main limitation of these methods. To address this problem, the proposed RANN is sensitive to the variation of data reliability across individual dimensions of every data sample.

Artificial Neural Network (ANN) [Hagan *et al.*, 1996] and its extensions [Specht, 1990; Sainath *et al.*, 2013; Yu *et al.*, 2013; Zhang and Woodland, 2015] are popular choices for accomplishing supervised learning tasks. However, the majority of existing neural network-based models is not able to handle unreliable input data. The RANN proposed in this paper as a novel type of neural network recognizes a reliability score for each dimension of each input multidimensional data point and also propagates such reliabilities during its inference procedure to yield the final output. Compared with traditional neural networks, the effects cast by the weights of links connecting pairs of neurons are automatically modulated based on reliability scores of information flowing through the corresponding links in the network, which are either directly supplied to the network by the input data or inferred on the fly during the network computing process.

Various approaches for assessing data reliability have been proposed recently [Pipino *et al.*, 2002; Cappiello *et al.*, 2004; Batini *et al.*, 2009]. For example, Pipino et al. [2002] used simple functional metrics such as *Simple Ratio, Min, Max,* and *Weighted Average* to assess data reliability. However, those simple metrics are not adaptable or sufficient to deal with the diverse types of complex data commonly seen today. In [Batini *et al.*, 2009], a wide range of techniques for assessing the quality of data is summarized. However those techniques apply classical statistical analysis techniques to assess data reliability by assuming all data are generated from a single source, which cannot be readily applied to deal with online user generated content due to the large number of authors involved as the multiple generation sources of the text. Cappiello et al. [2004] assessed the reliability of data from multiple users simultaneously by modeling user experiences dynamically captured during a person's software operation activities. Their method however cannot be applied to mine unreliable user generated content in a generic setting. To the best of our knowledge, none of the prior studies has comprehensively and systematically explored the issue of assessing data reliability of online user generated content for a large number of users simultaneously and subsequently making use of the assessed reliabilities to reason with uncertain data involving these user generated content. To fill the gap, this study examines the performance of the proposed RANN by applying it to mine a large volume of online user reviews for predicting product sales ranks and box office gross revenues as two demonstration cases. We choose online user reviews as unreliable multidimensional data in our experiments because of the high value of these reviews in revealing consumer opinions and purchase interests [Hu *et al.*, 2008].

3 Proposed Method

3.1 Problem Statement

Let $D = \{r^1, r^2, \ldots, r^m\}$ be a training dataset that comprises m independent labeled samples. For each sample $r^t = (\mathbf{x}^t, \mathbf{p}^t, \mathbf{y}^t)$ where $t \in [1, m]$, $\mathbf{x}^t = (x_1^t, x_2^t, \ldots, x_d^t)$ is a d-dimensional input vector; $\mathbf{p}^t = (p_1^t, p_2^t, \ldots, p_d^t)$ is a d-dimensional accompanying vector that represents the reliability of individual input components in $\mathbf{x}^t$; $\mathbf{y}^t = (y_1^t, y_2^t, \ldots, y_v^t)$ is a v-dimensional output vector. In $\mathbf{p}^t$, each $p_i^t \in [0, 1]$ $(1 \leq i \leq d)$ denotes the *reliability* of the input component x_i^t in $\mathbf{x}^t$ (the larger p_i^t is, the more reliable x_i^t is considered). For the convenience of mathematical deduction, we additionally introduce a d-dimensional vector $\mathbf{h}^t = (h_1^t, h_2^t, \ldots, h_d^t)$ that represents the *negative logarithmic reliability* of the input vector $\mathbf{x}^t$ where each h_i^t $(1 \leq i \leq d)$ is derived from p_i^t as follows:

$$h_i^t \triangleq \begin{cases} -\lg p_i^t & \text{if } \tau \leq p_i^t \leq 1 \\ -\lg \tau & \text{if } 0 \leq p_i^t < \tau \end{cases}, \tag{1}$$

where τ is a small positive number to avoid inputting a zero number to a log function.

In a multi-layered RANN, we let: L be the number of layers in the network, $\mathcal{N}_{i,l}$ be the i-th neuron lying on the l-th layer of the network, $w_{i,j}^l$ be the weight of the link connecting the neuron $\mathcal{N}_{i,l-1}$ to the neuron $\mathcal{N}_{j,l}$ where $2 \leq l \leq L$, b_j^l and c_j^l respectively be the *value bias* and the *reliability bias* of the neuron $\mathcal{N}_{j,l}$, χ^l be the set of neurons lying on the l-th layer of the network. Given a training sample r^t and a neuron $\mathcal{N}_{j,l}$, $z_j^{l,t}$, $r_j^{l,t}$, $o_j^{l,t}$, and $s_j^{l,t}$ respectively represent the neuron's *aggregate input value*, *aggregate input reliability*,

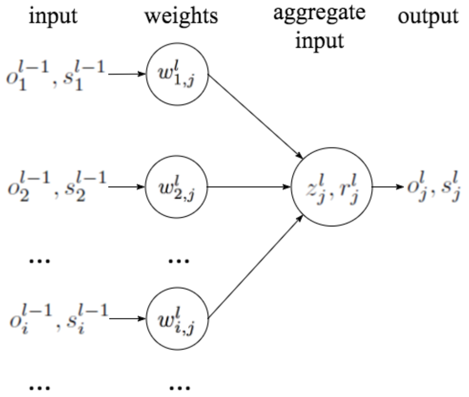


Figure 1: An example neuron in the proposed RANN.

output value, and *output reliability* as induced by the training sample. Fig. 1 shows a schematic diagram illustrating an example neuron in the proposed RANN where we model the relationships between these input and output variables as follows:

$$z_j^{l,t} \triangleq \sum_{i \in \chi^{l-1}} w_{i,j}^l \varphi(s_i^{l-1,t}) o_i^{l-1,t} + b_j^l; \tag{2}$$

$$r_j^{l,t} \triangleq \sum_{i \in \chi^{l-1}} w_{i,j}^l s_i^{l-1,t} + c_j^l; \tag{3}$$

$$o_j^{l,t} \triangleq \frac{1}{1 + \beta e^{-\alpha z_j^{l,t}}}; \tag{4}$$

$$s_j^{l,t} \triangleq \gamma r_j^{l,t} + \theta. \tag{5}$$

From the above equations we can see that both the input values and reliabilities fed into an RANN can be transformed to derive output values and their corresponding reliabilities. The weight of a link connecting a pair of neurons in an RANN can be modulated based on the reliability of information flowing through the link via the function $\varphi(\cdot)$ as indicated in Eq. (2). $\varphi(\cdot)$ can be implemented in at least the following three ways:

$$\varphi_1(x) \triangleq \frac{1}{1 + \lambda x}; \tag{6}$$

$$\varphi_2(x) \triangleq e^{-\lambda x}; \tag{7}$$

$$\varphi_3(x) \triangleq \frac{2e^{-\lambda x}}{1 + e^{-\lambda x}}. \tag{8}$$

Next we define E as an error function for the entire multi-layered RANN with respect to a given training dataset D as follows:

$$E \triangleq \frac{1}{2m} \sum_{t=1}^{m} \sum_{k \in \chi^L} (o_k^{L,t} - y_k^t)^2. \tag{9}$$

3.2 An Extended Backpropagation Algorithm for Training an RANN

The task of training an RANN requires learning the optimal values of the weights and biases of all neurons involved in the RANN, i.e. $\mathbf{w} = \{w_{i,j}^l\}$, $\mathbf{b} = \{b_j^l\}$, and $\mathbf{c} = \{c_j^l\}$, as well as the model configuration parameters for the RANN, i.e. α, β, γ, and θ, to minimize the error function E with respect to a given training dataset. We perform this learning task through a modified backpropagation algorithm, which is specially designed for the proposed RANN, by extending the classic backpropagation algorithm used in training a traditional neural network. For this training purpose, given a neural $\mathcal{N}_{j,l}$ in an RANN, we respectively define the neuron's *value error*, δ_j^l, and *reliability error*, ζ_j^l, as follows:

$$\delta_j^l = \frac{\partial E}{\partial z_j^l}; \quad \zeta_j^l = \frac{\partial E}{\partial r_j^l}. \tag{10}$$

Algorithm 1 specifies the aforementioned extended backpropagation algorithm for training an RANN, in which η is the gradient descent step size, which controls the learning speed. In the following, we give out the deduction of a few key steps in the algorithm. For simplicity, we omit the superscript t from all variable notations when it is clear that the only sample involved in the current training operation is r^t.

Deduction of Eq. (11):

$$\begin{aligned} \delta_k^L &= \frac{\partial E}{\partial z_k^L} = \partial \frac{1}{2m} \sum_{i \in \chi^L} (o_i^L - y_i)^2 / \partial z_k^L \\ &= \partial \frac{1}{2m} (o_k^L - y_k)^2 / \partial z_k^L = \frac{1}{m} (o_k^L - y_k) \frac{\partial o_k^L}{\partial z_k^L} \\ &= \frac{\alpha}{m} (o_k^L - y_k) o_k^L (1 - o_k^L). \end{aligned}$$

Deduction of Eq. (12):

$$\begin{aligned} \zeta_j^{L-1} &= \frac{\partial E}{\partial r_j^{L-1}} = \sum_{k \in \chi^L} \frac{\partial E}{\partial z_k^L} \frac{\partial z_k^L}{\partial s_j^{L-1}} \frac{\partial s_j^{L-1}}{\partial r_j^{L-1}} \\ &= \gamma \sum_{k \in \chi^L} \delta_k^L w_{jk}^L o_j^{L-1} \varphi'(s_j^{L-1}) \end{aligned}$$

Deduction of Eq. (13):

$$\begin{aligned} \delta_j^l &= \frac{\partial E}{\partial z_j^l} = \sum_{k \in \chi^{l+1}} \frac{\partial E}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} \\ &= \sum_{k \in \chi^{l+1}} \delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial o_j^l} \frac{\partial o_j^l}{\partial z_j^l} \\ &= \alpha \sum_{k \in \chi^{l+1}} \delta_k^{l+1} w_{jk}^{l+1} \varphi(s_j^l) o_j^l (1 - o_j^l). \end{aligned}$$

Deduction of Eq. (14):

$$\begin{aligned} \zeta_j^l &= \frac{\partial E}{\partial r_j^l} = \sum_{k \in \chi^{l+1}} \left(\frac{\partial E}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial r_j^l} + \frac{\partial E}{\partial r_k^{l+1}} \frac{\partial r_k^{l+1}}{\partial r_j^l} \right) \\ &= \sum_{k \in \chi^{l+1}} \left(\delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial s_j^l} \frac{\partial s_j^l}{\partial r_j^l} + \zeta_k^{l+1} \frac{\partial r_k^{l+1}}{\partial s_j^l} \frac{\partial s_j^l}{\partial r_j^l} \right) \\ &= \gamma \sum_{k \in \chi^{l+1}} w_{jk}^{l+1} \left(\delta_k^{l+1} o_j^l \varphi'(s_j^l) + \zeta_k^{l+1} \right) \end{aligned}$$

Algorithm 1 . Extended backpropagation algorithm for training an RANN.

```
Input: Training dataset $D = \{r^1, \ldots, r^m\}$, in which each
  $r^t = (\mathbf{x}^t, \mathbf{h}^t, \mathbf{y}^t)$.
Output: $\mathbf{w}, \mathbf{b}, \mathbf{c}$.
  Initialize:
  for $l = L, L-1, \ldots, 2$ do
      $\mathbf{w}_{ij}^l, \mathbf{b}_j^l, \mathbf{c}_j^l \leftarrow (1, \ldots, 1)$.
  end for
  for $t = 1, 2, \ldots, m$ do
      $\mathbf{o}^{1,t} = \mathbf{x}^t, \mathbf{s}^{1,t} = \mathbf{h}^t$.
      Feedforward:
      for $l = 2, 3, \ldots, L$ do
          Compute $\mathbf{z}^{l,t}$, $\mathbf{r}^{l,t}$, $\mathbf{o}^{l,t}$, $\mathbf{s}^{l,t}$ using Eqs. (2)-(8).
      end for
      Compute output error:
      for $k \in \chi^L$ do
          $\delta_k^{L,t} = \alpha(o_k^{L,t} - y_k^t) o_k^{L,t}(1 - o_k^{L,t}).$    (11)
          $\zeta_j^{L-1,t} = \gamma \sum_{k \in \chi^L} \delta_k^L w_{jk}^L o_j^{L-1} \varphi'(s_j^{L-1}).$    (12)
      end for
      Backpropagate the error:
      for $l = L-1, L-2, \ldots, 2$ do
          for $j \in \chi^l$ do
              $\delta_j^{l,t} = \alpha \sum_{k \in \chi^{l+1}} \delta_k^{l+1} w_{jk}^{l+1} \varphi(s_j^l) o_j^l (1 - o_j^l).$    (13)
              $\zeta_j^{l,t} = \gamma \sum_{k \in \chi^{l+1}} w_{jk}^{l+1} \big(\delta_k^{l+1} o_j^l \varphi'(s_j^l) + \zeta_k^{l+1}\big).$    (14)
          end for
      end for
  end for
  Weight update:
  for $l = L, L-1, \ldots, 2$ do
      for $j \in \chi^l$ do
          for $i \in \chi^{l-1}$ do
              $w_{i,j}^l \leftarrow w_{i,j}^l - \frac{\eta}{m} \sum_{t=1}^{m} \frac{\partial E}{\partial w_{i,j}^l}$
              $= w_{i,j}^l - \frac{\eta}{m} \sum_{t=1}^{m} \big(\delta_j^l o_i^{l-1} \varphi(s_i^{l-1}) + \zeta_j^l s_i^{l-1}\big);$
              $b_j^l \leftarrow b_j^l - \frac{\eta}{m} \sum_{t=1}^{m} \frac{\partial E}{\partial b_j^l} = b_j^l - \frac{\eta}{m} \sum_{t=1}^{m} \delta_j^{l,t};$    (15)
              $c_j^l \leftarrow c_j^l - \frac{\eta}{m} \sum_{t=1}^{m} \frac{\partial E}{\partial c_j^l} = c_j^l - \frac{\eta}{m} \sum_{t=1}^{m} \zeta_j^{l,t}.$
          end for
      end for
  end for
```

Deduction of Eq. (15):

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ij}^l} &= \frac{\partial E}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{ij}^l} + \frac{\partial E}{\partial r_j^l} \frac{\partial r_j^l}{\partial w_{ij}^l} \\
&= \delta_j^l o_i^{l-1} \varphi(s_i^{l-1}) + \zeta_j^l s_i^{l-1}; \\
\frac{\partial E}{\partial b_j^l} &= \frac{\partial E}{\partial z_j^l} \frac{\partial z_j^l}{\partial b_j^l} + \frac{\partial E}{\partial r_j^l} \frac{\partial r_j^l}{\partial b_j^l} = \delta_j^l; \\
\frac{\partial E}{\partial c_j^l} &= \frac{\partial E}{\partial z_j^l} \frac{\partial z_j^l}{\partial c_j^l} + \frac{\partial E}{\partial r_j^l} \frac{\partial r_j^l}{\partial c_j^l} = \zeta_j^l.
\end{aligned}
$$

4 Experimentation

4.1 Datasets

To explore the performance of the proposed RANN, we conduct a series of evaluation experiments using the following two popular datasets, both of which contain unreliable multi-dimensional data.

- *Amazon Web-data* [Leskovec and Krevl, 2014] contains both the product information and the corresponding user reviews of a product as collected on the Amazon site. In the experiment performed using the dataset, we use product features that specify a product's quantifiable attributes as well as user review ratings to predict the sales rank of a product among its product category. For example, the product features of a laptop as listed on Amazon include its model number and parameters regarding its processor, hard drive, display, and RAM. The groundtruth data regarding a product's sales rank is officially released by Amazon according to how well the product is sold overall among its product category or subcategory. Our experiment specifically examines two subcategories of products, namely *Laptops* and *Cellphones*, both under the category of *Electronics & Computers*. The two sub-datasets are denoted as "AMA-LAP" and "AMA-CEL" respectively. We choose to examine these two products in this study because of the detailed information associated with as well as the popularity and subjectivity of user reviews regarding both products.

- *Internet Movie Database (IMDb)* [IMDb, 2016] is a large collection of movie information, including detailed characteristics of a movie as well as its user reviews. Using this dataset, we aim to predict a movie's box office gross revenues through its detailed characteristics as well as the associated user ratings. Since the gross revenue of a movie may keep growing for a period of time, as well as the accumulation of its user ratings, we predict a movie's gross revenue at two timestamps, i.e. after the first week of its initial public showing and the latest moment as of the end of Jan. 2016. We use a million US dollar as the unit when recording a movie's gross revenue. This dataset is referred to as "IMDb" in the discussions in the rest of this session.

Table 1 lists the detailed information about the two datasets described above.

Table 1: Details of the experimental datasets used in this study.

AMA-LAP	
Number of products	9,325
Number of user ratings	141,782
Number of product features	25
Prediction target variable	Sales rank
Product features: operating system version, screen (size, resolution), processor (brand, model, count, speed), graphics coprocessor (brand, model), RAM (model, size, speed), hard drive (type, size), wireless, USB port (USB2.0, USB 3.0), brand, series, model, weight, dimension, color, battery, price.	
AMA-CEL	
Number of products	6,568
Number of user ratings	78930
Number of product features	10
Prediction target variable	Sales rank
Product features: networks, display, cameras, memory, processor, operating system version, battery, dimension, weight, price.	
IMDb	
Number of movies	7,288
Number of user ratings	345,381
Number of movie features	19
Prediction target variable	Gross revenue
Movie features: director, writer, casts, genre, keywords, motion picture rating, official sites, country, language, release date, filming location, budget, product Co., runtime, sound mix, color, aspect ratio, camera, format.	

4.2 Assessing Reliability of a Data Dimension

We regard the detailed information about a product or a movie as specified in our experimental datasets as reliable features since such information is typically provided by producers. Therefore, we set the *reliability* of each such feature as 1. In contrast, we regard the average user rating of a product or a movie as an unreliable feature due to the unreliable nature of online user ratings. The user ratings in both the Amazon and IMDb datasets are real numbers, whose value ranges are $[0, 5]$ and $[0, 10]$ respectively. For each type of user rating data, the proportion of online users who find the rating data helpful is additionally provided by the dataset curators, which is also leveraged to assess the *reliability* of average user ratings during the prediction tasks.

For each product/movie, we let $AvgRa$ be the average user rating of the product/movie; $RelRa$ be the reliability of the average user rating; Ra_i be the user rating provided by the i-th user; $StdRa$ be the standard deviation of user ratings for this product/movie; Pu_i be the number of people who find the rating Ra_i useful; Pr_i be the number of people who view the user rating Ra_i. Based on the above information, we estimate $RelRa$ according to the following equations:

$$\begin{aligned} AvgRa' &\triangleq \frac{\sum_i Ra_i Pu_i^2/Pr_i}{\sum_i Pu_i^2/Pr_i}, \\ z &\triangleq \frac{|AvgRa - AvgRa'|}{StdRa}, \\ RelRa &\triangleq P(|x| > z), \quad x \sim N(0,1). \end{aligned} \tag{16}$$

When designing the above equations for assessing the reliability of a data dimension in an input sample, we assume that the larger the difference between the average rating and the weighted average rating is, the more unreliable that the average rating would be.

4.3 Peer Methods

The following methods are adopted as peer methods to study the performance of the proposed RANN.

- Three baseline methods, including:
1) SVM—Support Vector Machine [Cortes and Vapnik, 1995].
2) ANN—Artificial Neural Network [Specht, 1991].
3) PNN— Probabilistic Neural Network [Specht, 1990]

- Two state-of-the-art methods for mining unreliable data, including:
1) SVMU—Support vector machine with input data uncertainty. [Zhang, 2005]
2) DTU—Decision tree for uncertain data. [Tsang *et al.*, 2011]

- Three versions of the proposed RANN model, including:
RANN-1—RANN implemented using Eq. (6);
RANN-2—RANN implemented using Eq. (7);
RANN-3—RANN implemented using Eq. (8).

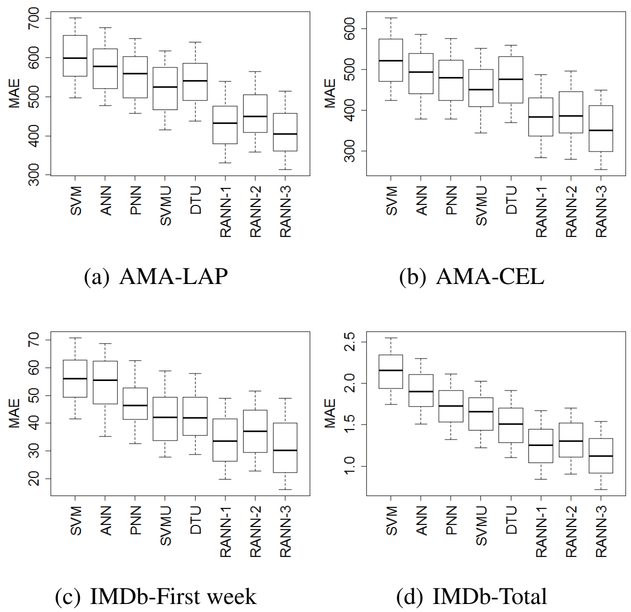
For the methods of SVM and SVMU, our implementation employs the LIBSVM package [Chang and Lin, 2011] and uses a Radial Basis Function (RBF) kernel wherein the configuration parameters are set to be $\{C = 1000, \gamma = 0.04\}$. For the method of DTU, our implementation employs the C50 package in R [Kuhn *et al.*, 2014]. For all neural network-based methods, namely ANN, PNN, RANN-1, RANN-2 and RANN-3, our implementation employs the RSNNS package in R [Bergmeir and Benítez Sánchez, 2012]. We adopt the multi-layer perceptron module and empirically optimize the gradient descent step size as $\eta = 0.2$. For SVMU and DTU, we modify the basic functions provided by the above packages according to the improvement procedures proposed in [Zhang, 2005; Tsang *et al.*, 2011] for better learning performance.

4.4 Evaluation Metrics

We employ two standard evaluation metrics to measure the performance of the proposed RANN as well as the series of aforementioned peer methods in respectively tackling the two prediction tasks. The two metrics include the *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)* [Li *et al.*, 2014], which can be computed as follows:

$$\begin{aligned} MAE &= \frac{\sum_{t=1}^{m} |O^{L,t} - y^t|}{m}; \\ RMSE &= \sqrt{\frac{\sum_{t=1}^{m} (O^{L,t} - y^t)^2}{m}}. \end{aligned} \tag{17}$$

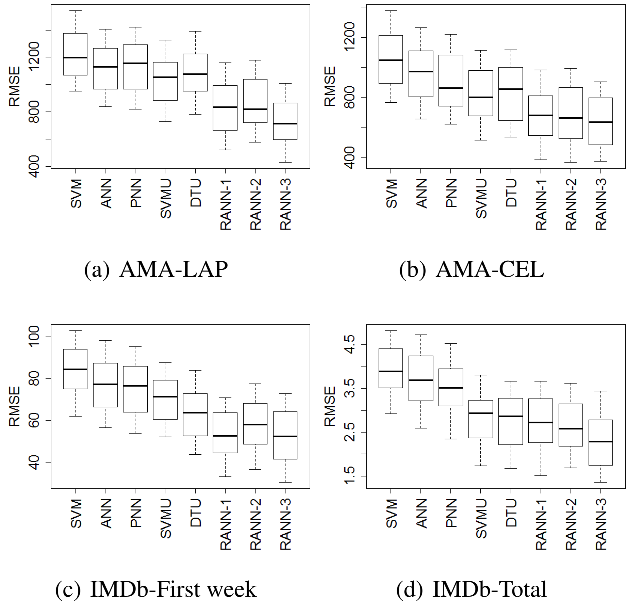
Please note that the dataset $D = \{r^1, r^2, \ldots, r^m\}$ used here is the testing dataset. In all our experiments, we use ten-fold



(a) AMA-LAP (b) AMA-CEL

(c) IMDb-First week (d) IMDb-Total

Figure 2: Performance comparison among RANN and peer methods in terms of MAE on different experimental datasets.



(a) AMA-LAP (b) AMA-CEL

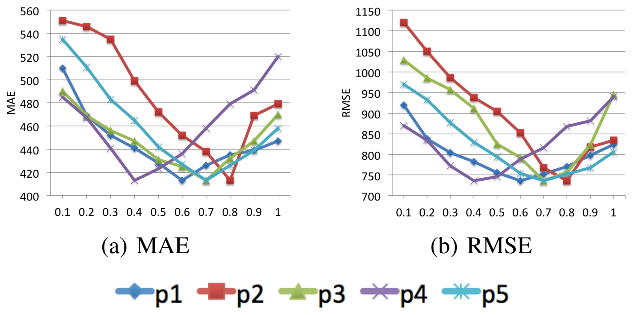
(c) IMDb-First week (d) IMDb-Total

Figure 3: Performance comparison among RANN and peer methods in terms of RMSE.

cross validation to carefully separate the use of data samples for the training versus testing purpose.

4.5 Results

Figs. 2 and 3 show the results of performance comparison among the proposed RANN and the aforementioned peer methods in tackling the two prediction tasks. From these results we can see that the third implementation of the proposed method, i.e. RANN-3, consistently outperforms all other methods in terms of both MAE and RMSE across all experiments performed. We also conduct independent t-tests to examine whether the proposed method significantly outperforms each of the peer methods. Results show that for every pair of RANN-3 and a peer method, including either of the two alternative implementation RANN-1 and RANN-2, the corresponding p-value is consistently below 0.05 for experiments performed over each dataset, which demonstrates the performance superiority of the proposed RANN-3 with respect to the rest of the methods with statistic significance. Due to space limit, we do not present these results here.



(a) MAE (b) RMSE

Figure 4: Performance variation of RANN under different configuration parameters.

The proposed RANN model carries five configuration parameters, namely $\alpha, \beta, \gamma, \theta$, and λ. To test the effects of these parameters on the performance of the proposed model, we additionally conduct a set of performance tuning experiments. Due to the space limit, we only report the performance of the third implementation of RANN, i.e. RANN-3, when it is applied onto the AMA-LAP dataset. The results are shown in Fig. 4, in which curves labeled with p_1 to p_5 respectively represents $\alpha, \beta, \gamma, \theta$, and λ. From these results we can see that the optimal value assignment for these configuration parameters are $(\alpha, \beta, \gamma, \theta, \lambda) = (0.6, 0.8, 0.7, 0.4, 0.7)$. Performance tuning experiments performed using other datasets yield similar optimal value assignment over these parameters, the results of which are not reported due to space limit.

5 Conclusion

In this paper, we propose a novel *Reliability-Aware Neural Network* (RANN) by extending the traditional artificial neural network for mining unreliable multidimensional data. The RANN is well suited for performing supervised learning tasks involving a large volume of online user-generated data, such as online reviews. The new RANN is also equipped with a pre-processing step that assesses the reliability of each dimension of each input data sample that is aggregated from a volume of unreliable raw data, such as online user reviews. Experimental results demonstrate the effectiveness of the proposed RANN through two application cases, one regarding predicting product sales ranks and the other involving predicting box office gross revenues. The proposed RANN consistently outperforms a series of peer methods and state-of-the-art algorithms in these prediction tasks, the advantage of which is clearly demonstrated through all experimental results.

References

[Aggarwal and Yu, 2009] Charu C Aggarwal and Philip S Yu. A survey of uncertain data algorithms and applications. *Knowledge and Data Engineering, IEEE Transactions on*, 21(5):609–623, 2009.

[Aggarwal, 2010] Charu C Aggarwal. *Managing and mining uncertain data*, volume 35. Springer Science & Business Media, 2010.

[Batini *et al.*, 2009] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM computing surveys (CSUR)*, 41(3):16, 2009.

[Bergmeir and Benítez Sánchez, 2012] Cristoph Norbert Bergmeir and José Manuel Benítez Sánchez. Neural networks in r using the stuttgart neural network simulator: Rsnns. American Statistical Association, 2012.

[Cappiello *et al.*, 2004] Cinzia Cappiello, Chiara Francalanci, and Barbara Pernici. Data quality assessment from the user's perspective. In *Proceedings of the 2004 international workshop on Information quality in information systems*, pages 68–73. ACM, 2004.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[Chui and Kao, 2008] Chun-Kit Chui and Ben Kao. A decremental approach for mining frequent itemsets from uncertain data. In *Advances in Knowledge Discovery and Data Mining*, pages 64–75. Springer, 2008.

[Chui *et al.*, 2007] Chun-Kit Chui, Ben Kao, and Edward Hung. Mining frequent itemsets from uncertain data. In *Advances in knowledge discovery and data mining*, pages 47–58. Springer, 2007.

[Cormode and McGregor, 2008] Graham Cormode and Andrew McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 191–200. ACM, 2008.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[Dalvi and Suciu, 2007] Nilesh Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4):523–544, 2007.

[Hagan *et al.*, 1996] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996.

[Hu *et al.*, 2008] Nan Hu, Ling Liu, and Jie Jennifer Zhang. Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. *Information Technology and Management*, 9(3):201–214, 2008.

[IMDb, 2016] IMDb. Internet movie database. http://www.imdb.com/, 2016. Last Accessed: 2016-02-01.

[Kuhn *et al.*, 2014] M Kuhn, S Weston, N Coulter, and R Quinlan. C50: C5. 0 decision trees and rule-based models. *R package version 0.1. 0-21, URL http://CRAN. R-project. org/package C*, 50, 2014.

[Lee *et al.*, 2007] Sau Dan Lee, Ben Kao, and Reynold Cheng. Reducing uk-means to k-means. In *ICDM Workshop 2007*, pages 483–488. IEEE, 2007.

[Leskovec and Krevl, 2014] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[Li *et al.*, 2014] Fangtao Li, Sheng Wang, Shenghua Liu, and Ming Zhang. Suit: A supervised user-item based topic model for sentiment analysis. In *AAAI 2014*, 2014.

[Minku and Yao, 2012] Leandro L Minku and Xin Yao. Using unreliable data for creating more reliable online learners. In *IJCNN 2012*, pages 1–8. IEEE, 2012.

[Pipino *et al.*, 2002] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[Qin *et al.*, 2009a] Biao Qin, Yuni Xia, and Fang Li. Dtu: a decision tree for uncertain data. In *Advances in Knowledge Discovery and Data Mining*, pages 4–15. Springer, 2009.

[Qin *et al.*, 2009b] Biao Qin, Yuni Xia, Sunil Prabhakar, and Yicheng Tu. A rule-based classification algorithm for uncertain data. In *ICDE 2009.*, pages 1633–1640. IEEE, 2009.

[Qin *et al.*, 2010] Xiangju Qin, Yang Zhang, Xue Li, and Yong Wang. Associative classifier for uncertain data. In *Web-Age Information Management*, pages 692–703. Springer, 2010.

[Sainath *et al.*, 2013] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for lvcsr. In *ICASSP 2013*, pages 8614–8618. IEEE, 2013.

[Specht, 1990] Donald F Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.

[Specht, 1991] Donald F Specht. A general regression neural network. *Neural Networks, IEEE Transactions on*, 2(6):568–576, 1991.

[Tsang *et al.*, 2011] Smith Tsang, Ben Kao, Kevin Y Yip, Wai-Shing Ho, and Sau Dan Lee. Decision trees for uncertain data. *Knowledge and Data Engineering, IEEE Transactions on*, 23(1):64–78, 2011.

[Yu *et al.*, 2013] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *ICASSP 2013*, pages 7893–7897. IEEE, 2013.

[Zhang and Woodland, 2015] Chao Zhang and Philip C Woodland. A general artificial neural network extension for htk. *Submission to InterSpeech*, 2015.

[Zhang, 2005] Jinbo Bi Tong Zhang. Support vector classification with input data uncertainty. *Advances in neural information processing systems*, 17:161, 2005.