GBC: Gradient Boosting Consensus Model for Heterogeneous Data

Xiaoxiao Shi* Jean-Francois Paiement† David Grangier‡ Philip S. Yu*§

Abstract

With the rapid development of database technologies, multiple data sources may be available for a given learning task (*e.g.*, collaborative filtering). However, the data sources may contain different types of features. For example, users' profiles can be used to build recommendation systems. In addition, a model can also use users' historical behaviors and social networks to infer users' interests on related products. We argue that it is desirable to collectively use any available multiple heterogeneous data sources in order to build effective learning models. We call this framework *heterogeneous learning*. In our proposed setting, data sources can include (i) non-overlapping features, (ii) non-overlapping instances, and (iii) multiple networks (i.e. graphs) that connect instances. In this paper, we propose a general optimization framework for heterogeneous learning, and devise a corresponding learning model from gradient boosting. The idea is to minimize the empirical loss with two constraints: (1) There should be consensus among the predictions of overlapping instances (if any) from different data sources; (2) Connected instances in graph datasets may have similar predictions. The objective function is solved by stochastic gradient boosting trees. Furthermore, a weighting strategy is designed to emphasize informative data sources, and deemphasize the noisy ones. We formally prove that the proposed strategy leads to a tighter error bound. This approach consistently outperforms a standard concatenation of data sources on movie rating prediction, number recognition and terrorist attack detection tasks. Furthermore, the approach is evaluated on a distributed database from a nationa wide phone provider with over 500,000 instances, 91 different data sources, and over 45,000,000 joined features. We observe that the proposed model can improve out-of-sample error rate substantially.

1 Introduction

With the rapid development of database technologies, multiple related data sources can be used to build prediction models given a target task. Each of the related data sources may have a distinct set of features and instances, and we argue that the combination of all data sources may yield better prediction results. An example is illustrated in Fig. 1. The task is to predict movie ratings

*Computer Science Department, University of Illinois at Chicago, USA. {xshi9, psyu}@uic.edu.

†AT&T Labs, USA. jpaiement@research.att.com.

‡Microsoft Research, USA. grangier@microsoft.com; D. Grangier was with AT&T Labs Research when this research has been performed

§Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia

in the Internet Movie Database (IMDB[1]), which has been used in movie recommendation [1]. For example, in Fig. 1(a), given that we observe that the rating for "The Godfather" is 9.2 (out of 10), and "The Giant Spider Invasion" is 2.8, what are the ratings for "Apocalypse Now" and "Monster a-Go Go"? Note that in this task, there are multiple available databases that record various information about movies. For instance, there is a genre database (Fig. 1(b)), a sound technique database (Fig. 1(c)), a running times database (Fig. 1(d)), an actor graph database that links two movies together if the same actor/actress performs in the movies (Fig. 1(e)), and a director graph database that links two movies if they are directed by the same director (Fig. 1(f)). Note that these multiple data sources have the following properties:

- Firstly, each data source can have its own feature sets. For example, the running times database (Fig. 1(d)) has numerical features; the genre database (Fig. 1(b)) has nominal features, and the actor graph database (Fig. 1(e)) provides graph relational features.
- Secondly, each data source can have its own set of instances. For example, the genre database does not have the record for "Monster a-Go Go"; the running times database does not have any record of "Apocalypse Now".

Note that it is difficult to build an accurate prediction model by using only one of the five databases, since the information in each of them is incomplete. However, if we consider the five data sources collectively, we are able to infer that the rating of "Apocalypse Now" (ground truth: 8.6) may be close to that of "The Godfather", since they are similar in genre and they are connected in the actor graph. Similarly, one can infer that the rating for "Monster a-Go Go" (ground truth: 1.5) is similar to that of "The Giant Spider Invasion".

In the past, multi-view learning [2, 3] was proposed to study a related problem where each instance can have different views. However, it usually does not consider graph data with relational features, especially when there are multiple graphs and each graph may only contain a subset of the relation features. Hence, we study a more general learning scenario called *heterogeneous learning* where the data can come from multiple sources. Specifically, the data sources can (1) have non-overlapping features (*i.e.*, new features in certain data sources), (2) have some non-overlapping instances (*i.e.*, new objects/instances in certain data sources), and (3) contain multiple network (i.e. weighted graphs) datasets. Furthermore, some of the data sources may contain substantial noise or low-quality data. Our aim is to utilize all data sources collectively and judiciously, in order to improve the learning performance.

A general objective function is proposed to make good use of the information from these multiple data sources. The intuition is to learn a prediction function from each data source to minimize the empirical loss with two constraints. First, if there are overlapping instances, the predictions of the same instance should be similar even when learning from different data sources. Second, the predictions of connected data (*i.e.*, instances connected in any of the graphs) should be similar. Finally, the prediction models are judiciously combined (with different weights) to generate a global prediction model. In order to solve the objective function, we borrow ideas from gradient boosting decision trees (GBDT), which is an iterated algorithm that generates a sequence of decision trees, where each tree fits the gradient residual of the objective function. We call our proposed algorithm Gradient Boosting Consensus (GBC) because each data source

[1] http://www.imdb.com/

Name	Ratings
The Godfather	9.2
Apocalypse Now	?
Monster a-Go Go	?
The Giant Spider Invasion	2.8

(a) Movie rating prediction.

Name	Genre
The Godfather	Drama, Crime
Apocalypse Now	Drama, War
The Giant Spider Invasion	Horror, Sci-Fi

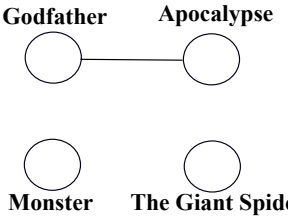
(b) Genre database.

Name	Sound Technique
Apocalypse Now	DTS, Digital, 6-Track
Monster a-Go Go	Mono
The Giant Spider Invasion	Mono

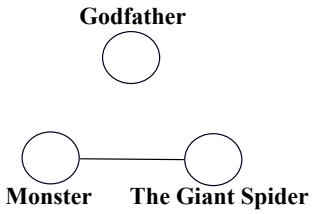
(c) Sound technique database.

Name	Running times (mins)
The Godfather	175
Monster a-Go Go	70
The Giant Spider Invasion	84

(d) Running times.



(e) Actor graph.



(f) Director graph that does not have record on "Apocalypse Now".

Figure 1: Combining different sources to infer movie ratings. The true rating for "Apocalypse Now" is 8.6, while the rating for "Monster a-Go Go" is 1.5.

generates a set of trees, and the consensus of the decision trees makes the final prediction. Moreover, GBC has the following properties.

- Deep-ensemble. Recall that the traditional boosting tree model is an iterated algorithm that builds new trees based on the previous iterations (residuals). Usually, these new trees are generated based on the residual of only one data source. However, as shown in Fig. 2, GBC generates new trees collectively from all data sources (horizontally) in each iteration (vertically). We call it "deep ensemble" since it ensembles models both horizontally and vertically to make the final prediction.

- Network-friendly. Unlike traditional boosting trees, GBC can take advantage of multiple graph datasets to improve learning. In other words, it can take advantage of traditional vector-based features and graph relational features simultaneously.

- Robust. Some data sources may contain substantial noise. A weighting strategy is incorporated into GBC to emphasize informative data sources and deemphasize the noisy ones. This weighting strategy is further proven to have a tighter error bound in both inductive and transductive settings.

We conducted four sets of experiments. These experiments include IMDB movie rating prediction, UCI number recognition, terrorist attack detection, and a demographic prediction

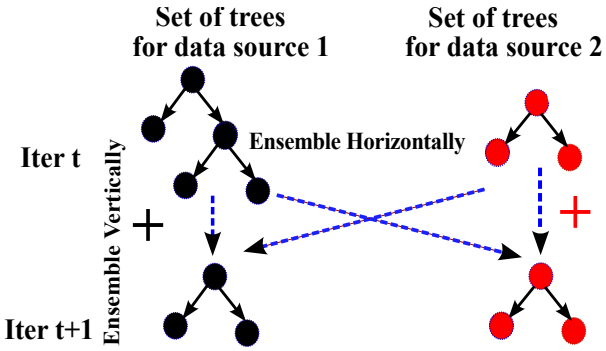


Figure 2: Gradient Boosting Consensus.

task in a big dataset from a nationa wide phone provider with over 500,000 samples, 91 different data sources, and over 45,000,000 joined features. Each task has a set of data sources with heterogeneous features. For example, in the IMDB movie rating prediction task, we have data sources about the plots of the movies (text data), technologies used by the movies (nominal features), running times of the movies (numerical features), and several movie graphs (such as director graph, actor graph). All these mixture types of data sources were used collectively to build a prediction model. Since there is no previous model that can handle the problem directly, we have constructed a straightforward baseline which first appends all data sources together into a single database, and uses traditional learning models to make predictions. Experiments show that the proposed GBC model consistently outperforms our baseline, and can decrease the error rate by as much as 80%.

2 Related Work

There are several areas of related works upon which our proposed model is built. First, multi-view learning (*e.g.*, [2, 4, 5]) is proposed to learn from instances which have multiple views in different feature spaces. For example, in [4], a co-training algorithm is proposed to classify the web pages by the text on the web page, and the text on hyperlinks pointing to the web page. In [5], a general clustering framework is proposed to reconcile the clustering results from different views. In [6], a term called consensus learning is proposed. The general idea is to perform learning on each heterogeneous feature space independently and then summarize the results via ensemble. Recently, [7] proposes a recommendation model (collaborative filtering) that can combine information from different contexts. It finds a latent factor that connects all data sources, and propagate information through the latent factor. There are mainly two differences between our work and the previous approaches. First, most of the previous works do not consider the vector-based features and the relational features simultaneously. Second and foremost, most of the previous works require the data sources to have records of all instances in order to enable the mapping, while the proposed GBC model does not have this constraint.

Another area of related work is collective classification (*e.g.*, [8]) that aims at predicting the class label from a network. Its key idea is to combine the supervision knowledge from traditional vector-based feature vectors, as well as the linkage information from the network. It has been applied to various applications such as part-of-speech tagging [9], classification of hypertext documents using hyperlinks [10], *etc.* Most of these works study the case when there

is only one vector-based feature space and only one relational feature space, and the focus is how to combine the two. Different from traditional collective classification framework, we consider multiple vector-based features and multiple relational features simultaneously. Specifically, [11] proposes an approach to combine multiple graphs to improve the learning. The basic idea is to average the predictions during training. There are three differences between the previous works and the current model. Firstly, we allow different data sources to have non-overlapping instances. Secondly, we introduce a weight learning process to filter out noisy data sources. Thirdly, we consider *multiple* vector-based sources and *multiple* graphs at the same time. Hence, all the aforementioned methods could not effectively learn from the datasets described in Section 4, as they all contain multiple vector-based data sources and relational graphs.

Another related field is transfer learning (e.g., [12, 13, 14, 15, 16]) which aims at learning the target task from a related out-of-domain source task. Most research work on transfer learning focus on how to make good use of the training data that distributes differently with the test data. A general approach is based on re-sampling (e.g., [12]), where the motivation of it is to "emphasize" the knowledge among "similar" and discriminating instances. Another line of research is to find a new feature space in which the training and test data have strong similarities (e.g., [17, 18, 19, 20, 21, 22, 23, 24]). For instance, [25] applies sparse learning techniques to transfer the knowledge across multiple tasks. There is also a set of works that enable transfer learning on document-related tasks. For example, [21] finds the commonality among different tasks, such as common words, to attack NLP tasks. However, documents are usually viewed as coming from the same feature space, since any document can be represented as "bag of words" where the dictionary contains all possible words. Different from these works, we do not require the original training and test datasets to be in the same feature space, or have a subset of common features. Instead, they can be from completely different feature spaces, and they can even be graph datasets.

The proposed model is also related to the research of ensemble learning. For example, gradient boosting tree [26] is a work proposed by J. H. Friedman to approach the learning objective by building an ensemble of weak learners (*i.e.*, decision tree in this case). It is achieved by fitting a decision tree to the residual error of the model at each iteration, and the final model is composed of a weighted summation of all the fitted trees. GBDT is a new model widely used in the industry. For instance, it is used in Yahoo and the search engine company Yandex in the field of "learning to rank" [27]; it is also adopted by the winning team of the Netflix competition [28]. In this paper, we borrow the idea of gradient boosting to solve the problem of aggregating multiple heterogeneous data sources. There are mainly two reasons that we choose this technique. First, in heterogeneous learning, ensemble approach is a natural choice since it is not straightforward to come up with a single model to deal with multiple heterogeneous data sources. Second, it is required that the prediction model to be efficient since large dataset is used as input. As such, gradient boosting tree is a clear option, especially given that it can be easily deployed in a distributed computing environment. Note that there are some other ensemble learning researches developed with the idea of consensus. For instance, [29] is proposed to perform clustering via an ensemble learning approach. The basic idea is to model the pairwise relationships from multiple sources, and construct the "belief" graphs that maximizes the consensus among the data sources. Furthermore, a generalized unsupervised learning model is proposed in [30], which also adopts the idea of aggregating multiple data sources via consensus principle. However, so far as we

Table 1: Symbol definition

Symbol	Definition
$\mathbf{x}_j^{(i)} \in \mathbb{R}^{d_i}$	The j-th data (column vector) in the i-th source (the i-th feature space).
G_g	The g-th relational graph.
$\mathcal{U}_i$	The set of unlabeled data in the i-th data source.
$f_i(\mathbf{x})$	The prediction model built from the i-th data source.
$\mathcal{C}$	Consensus constraint.
$\mathcal{G}$	Graph connectivity constraint.
$\mathcal{T}$	Set of labeled data.

know, although there are several ensemble methods proposed to aggregate heterogeneous sources in the unsupervised learning framework (as in [29, 30]), there is only a few proposed to conduct supervised heterogeneous learning. Furthermore, the proposed model also borrows the idea from the research of multiple kernel learning [31]. The idea in MKL research is to use multiple kernels in formulating a learning process. In this way, the learning model itself will pick the best kernels to improve the result. In this paper, the proposed model attacks a more general but more challenging problem. It aims at choosing the optimal data sources, among which some of them have totally different statistics and data structures.

3 Problem Formulation

In this section, we formally define the problem of heterogeneous learning, and then introduce a general learning objective. In heterogeneous learning, data can be described in heterogeneous feature spaces from multiple sources. Traditional vector-based features are denoted with the column vectors $\mathbf{x}_i^{(j)} \in \mathbb{R}^{d_j}$ corresponding to the i-th data in the j-th source (or the j-th feature space) whose dimension is d_j. In matrix form, $\mathbf{X}^{(j)} = [\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \cdots, \mathbf{x}_m^{(j)}] \in \mathbb{R}^{d_j \times m}$ is the dataset in the j-th feature space where m is the sample size. Different from vector-based features, graph relational features describe the relationships between instances. In other words, they are graphs representing connectivity/similarity of the data. Specifically, we denote $G_g =< V_g, E_g >$ as the g-th graph where V_g is the set of nodes and $E_g \subseteq V_g \times V_g$ is the set of edges. We assume that the features from the same data source are from the same feature space, and hence each data source has a corresponding feature space. Furthermore, different data sources may provide different sets of instances. In other words, some instances exist in some data sources, but are missing in the others. Thus, heterogeneous learning is a machine learning scenario where we consider data from different sources, but they may (1) have different sets of instances, (2) have different feature spaces, and (3) have multiple network based (graph) datasets. Hence, we have p data sources providing vector-based features $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(p)}$ and q data sources providing relational networks $G_1, \cdots, G_q$. The aim is to derive learning models (classification, regression or clustering) by collectively and judiciously using the $p + q$ data sources. A set of important symbols in the remaining of the paper are summarized in Table 1.

4 Gradient Boosting Consensus

In this section, we describe the general framework of the proposed GBC model and its theoretical foundations.

4.1 The GBC framework In order to use multiple data sources, the objective function aims at minimizing the overall empirical loss in all data sources, with two more constraints. First, the overlapping instances should have similar predictions from the models trained on different data sources, and we call this the *principle of consensus*. Second, when graph relational data is provided, the connected data should have similar predictions, and we call this the *principle of connectivity similarity*. In summary, the objective function can be written as follows:

$$\begin{aligned} \min \mathcal{L} &= \sum_i w_i \sum_{\mathbf{x} \in \mathcal{T}} \mathbf{L}(f_i(\mathbf{x}), y) \\ \text{s.t. } \mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) &= 0 \\ \mathcal{G}\big(\mathbf{f}, \mathbf{w}\big) &= 0 \end{aligned} \tag{4.1}$$

where $\mathbf{L}(f_i(\mathbf{x}), y)$ is the empirical loss on the set of training data $\mathcal{T}$, w_i is the weight of importance of the i-th data source, which is discussed in Section 4.3. Furthermore, the two constraints $\mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) = 0$ and $\mathcal{G}\big(\mathbf{f}, \mathbf{w}\big) = 0$ are the two assumptions discussed above, which are the principle of consensus and principle of connectivity similarity, respectively. More specifically, the consensus constraint $\mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) = 0$ is defined as follows:

$$\begin{aligned} \mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) &= \sum_i w_i \sum_{\mathbf{x} \in \mathcal{U}_i} \mathbf{L}\Big(f_i(\mathbf{x}), \mathbb{E}\big(f(\mathbf{x})\big)\Big) \\ \mathbb{E}\big(f(\mathbf{x})\big) &= \sum_{\{i | \mathbf{x} \in \mathcal{U}_i\}} w_i f_i(\mathbf{x}) \\ \text{s.t. } \sum_i w_i &= 1 \end{aligned} \tag{4.2}$$

It first calculates the expected prediction $\mathbb{E}\big(f(\mathbf{x})\big)$ of a given unlabeled instance $\mathbf{x}$, by summarizing the current predictions from multiple data sources $\sum_{\{i|\mathbf{x} \in \mathcal{U}_i\}} w_i f_i(\mathbf{x})$. This expectation is computed only from the data sources that contain $\mathbf{x}$; in other words, it is from the data sources whose indices are in the set $\{i | \mathbf{x} \in \mathcal{U}_i\}$ where $\mathcal{U}_i$ is the set of unlabeled instances in the i-th data source. Hence, if the j-th data source does not have record of $\mathbf{x}$, it will not be used to calculate the expected prediction. This strategy enables GBC to handle non-overlapping instances in multiple data sources, and uses overlapping instances to improve the consensus. Eq. 4.2 forces the predictions of $\mathbf{x}$ (*e.g.*, $f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots$) to be close to $\mathbb{E}\big(f(\mathbf{x})\big)$.

Furthermore, according to the principle of connectivity similarity, we introduce another constraint $\mathcal{G}\big(\mathbf{f}, \mathbf{w}\big)$ as follows:

$$\begin{aligned} \mathcal{G}\big(\mathbf{f}, \mathbf{w}\big) &= \sum_i w_i \sum_{\mathbf{x} \in \mathcal{U}_i} \mathbf{L}\big(f_i(\mathbf{x}), \tilde{\mathbb{E}}_i\big(f(\mathbf{x})\big) \\ \tilde{\mathbb{E}}_i\big(f(\mathbf{x})\big) &= \sum_g \frac{\hat{w}_g}{|\{(z, x) \in G_g\}|} \sum_{(\mathbf{z}, \mathbf{x}) \in G_g} f_i(\mathbf{z}) \\ \text{s.t. } \sum_g \hat{w}_g &= 1 \end{aligned} \tag{4.3}$$

The above constraint encourages connected data to have similar predictions. It works by calculating the graph-based expected prediction of $\mathbf{x}$ by looking at the average prediction ($\frac{1}{|\{(z,x)\in G_g\}|}\sum_{(\mathbf{z},\mathbf{x})\in G_g} f_i(\mathbf{z})$) of all its connected neighbors ($\mathbf{z}$'s). If there are multiple graphs, all the expected predictions are summarized by the weights $\hat{w}_g$.

We use the method of Lagrange multipliers [32] to solve the constraint optimization in Eq. 4.1. The objective function becomes

$$\min \mathcal{L} = \sum_i w_i \sum_{\mathbf{x}\in\mathcal{T}} \mathbf{L}(f_i(\mathbf{x}), y) + \lambda_0 \mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) + \lambda_1 \mathcal{G}\big(\mathbf{f}, \mathbf{w}\big) \tag{4.4}$$

where the two constraints $\mathcal{C}\big(\mathbf{f}, \mathbf{w}\big)$ and $\mathcal{G}\big(\mathbf{f}, \mathbf{w}\big)$ are regularized by Lagrange multipliers λ_0 and λ_1. These parameters are determined by cross-validation, which is detailed in Section 5. Note that in Eq. 4.4, the weights w_i and $\hat{w}_g$ $(i, g = 1, 2, \cdots)$ are essential. On one hand, the w_is are introduced to assign different weights to different vector-based data sources. Intuitively, if the t-th data source is more informative, w_t should be large. On the other hand, the $\hat{w}_g$s are the weights for the graph relational data sources. Similarly, the aim is to give high weights to important graph data sources, while deemphasizing the noisy ones. We define different weight symbols (w_i and $\hat{w}_g$) for the data sources with vector-based features (w_i) and graph relational features ($\hat{w}_g$). The values of the weights are automatically learned and updated in the training process, as discussed in Section 4.3.

4.2 Model training of GBC We use stochastic gradient descent [26] to solve the optimization problem in Eq. 4.4. In general, it is an iterated algorithm that updates the prediction functions $\mathbf{f}(\mathbf{x})$ in the following way:

$$\mathbf{f}(\mathbf{x}) \leftarrow \mathbf{f}(\mathbf{x}) - \rho \frac{\partial \mathcal{L}}{\partial \mathbf{f}(\mathbf{x})}$$

It is updated iteratively until a convergence condition is satisfied. Specifically, inspired by gradient boosting decision trees (or GBDT [26]), a regression tree is built to fit the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{f}(\mathbf{x})}$, and the best parameter ρ is explored via line search [26]. Note that the calculation of $\frac{\partial \mathcal{L}}{\partial \mathbf{f}(\mathbf{x})}$ depends on the loss function $\mathbf{L}(f, y)$ as reflected in Eq. 4.1. In the following, we use the L-2 loss (for regression problems) and the binary logistic loss (for binary classification problem) as examples:

GBC with L-2 Loss: In order to update the prediction function of the i-th data source, we follow the gradient descent formula as follows.

$$f_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) - \rho \frac{\partial \mathcal{L}}{\partial f_i(\mathbf{x})} \tag{4.5}$$

If the L-2 loss is used in $\mathcal{L}$, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial f_i(\mathbf{x})} =& 2w_i\Big(\sum_{\mathbf{x}\in\mathcal{T}}(f_i(\mathbf{x}) - y) + \lambda_0 \sum_{\mathbf{x}\in\mathcal{U}}(f_i(\mathbf{x}) - \mathbb{E}) \\ &+ \lambda_1 \sum_{\mathbf{x}\in\mathcal{U}}(f_i(\mathbf{x}) - \tilde{\mathbb{E}}_i)\Big)\end{aligned}$$

The L-2 loss is a straightforward loss function for the GBC model, and it is used to perform regression tasks in Section 5.

GBC with Logistic Loss: With logistic loss, the partial derivative in Eq. 4.5 becomes:

$$\frac{\partial \mathcal{L}}{\partial f_i(\mathbf{x})} = w_i \Big(\sum_{\mathbf{x} \in \mathcal{T}} \frac{-y e^{-y f_i(\mathbf{x})}}{1 + e^{-y f_i(\mathbf{x})}} + \lambda_0 \sum_{\mathbf{x} \in \mathcal{U}} \frac{-\mathbb{E} e^{-\mathbb{E} f_i(\mathbf{x})}}{1 + e^{-\mathbb{E} f_i(\mathbf{x})}} + \lambda_1 \sum_{\mathbf{x} \in \mathcal{U}} \frac{-\tilde{\mathbb{E}} e^{-\tilde{\mathbb{E}} f_i(\mathbf{x})}}{1 + e^{-\tilde{\mathbb{E}} f_i(\mathbf{x})}} \Big)$$

Note that the above formula uses the binary logistic loss where $y = -1$ or $y = 1$, but one can easily extend this model to tackle multi-class problems by using the one-against-others strategy. In Section 5, we adopt this strategy to handle multi-class problems.

With the updating rule, we can build the GBC model as described in Algorithm 1. It first finds the initial prediction models for all data sources in Step 1. Then, it goes into the iteration (Step 3 to Step 11) that generates a series of decision trees. The basic idea is to follow the updating rule in Eq. 4.5, and build a decision tree $g_i(x^i)$ to fit the partial derivative of the loss (Step 5). Furthermore, we follow the idea of [26], and let the number of iterations N be set by users. In the experiment, it is determined by cross-validation.

Then given a new data $\mathbf{x}$, the predicted output is

$$\hat{f}(\mathbf{x}) = \mathbf{P}(\sum \omega_i \hat{f}_i(\mathbf{x}^i)) \tag{4.6}$$

where $\mathbf{P}(y)$ is a prediction generation function, where $\mathbf{P}(y) = y$ in regression problems, and $\mathbf{P}(y) = 1$ iff $y > 0$ ($\mathbf{P}(y) = -1$ otherwise) in binary classification problems.

4.3 Weight Learning In the objective function described in Eq. 4.4, one important element is the set of weights (w_i and $\hat{w}_g$) for the data sources. Ideally, informative data sources will have high weights, and noisy data sources will have low weights. As such, the proposed GBC model can judiciously filter out the data sources that are noisy. To this aim, we design the weights by looking at the empirical loss of the model trained from the data source. Specifically, if a data source induces large loss, its weight should be low. Following this intuition, we design the weight as follows:

$$w_i = \exp\Big(- \sum_{\mathbf{x} \in \mathcal{L}} \mathbf{L}\big(f_i(\mathbf{x}), y\big)/z \Big) \tag{4.10}$$

where $\mathbf{L}\big(f_i(\mathbf{x}), y\big)$ is the empirical loss of the model trained from the i-th data source, and z is a normalization constant to ensure the summation of w_is equals to one. Note that the definition of the weight w_i is derived from the weighting matrix in normalized cut [33]. The exponential part can effectively give penalty to large loss. Hence, w_i will be large if the empirical loss of the i-th data source is small; it becomes small if the loss is large. It is proven in Theorem 4.1 that the updating rule of the weights in Eq. 4.10 can result in a smaller error bound. Similarly, we define the weights for graph data sources as follows:

$$w_g = \exp\Big(- \frac{1}{c} \sum_{\mathbf{x}_a^g \sim \mathbf{x}_b^g} \sum_i w_i \mathbf{L}\big(f_i(\mathbf{x}_a), f_i(\mathbf{x}_b)\big)/z \Big) \tag{4.11}$$

Input: Data from different sources (including vector-based and graph data): $\mathcal{X}_1$, $\mathcal{X}_2$, $\cdots$, $\mathcal{X}_p$, Expected outputs (labels or regression values) of a subset of data $\mathcal{Y}$. Number of iterations N.

Output: The prediction model GBC $\hat{f}(\mathbf{x})$.

Initialize $\hat{f}_i(\mathbf{x})$ to be a constant such that $\hat{f}_i(\mathbf{x}) = \arg\min_{\rho_i} \sum_{\mathbf{x}\in\mathcal{L}} \mathbf{L}(\rho_i, y)$ for $i = 1, 2, \cdots, p$.

Initialize $w_i = \frac{1}{p}$.

for $t = 1, 2, \cdots, N$ **do**

  for $i = 1, 2, \cdots, p$ **do**

    For all $\mathbf{x}^{(i)}$, compute the negative gradient with respect to $f(\mathbf{x}^{(i)})$:

$$z_i = -\frac{\partial}{\partial f_i(\mathbf{x}^{(i)})} \mathcal{L}\big(\mathbf{f}(\mathbf{x}), \mathbf{w}\big) \tag{4.7}$$

    where $\mathcal{L}$ is defined in Eq. 4.4 with both vector-based and graph data.

    Fit a regression model $g_i(\mathbf{x}^{(i)})$ that predicts z_i's from $\mathbf{x}^{(i)}$'s.

    Line search to find the optimal gradient descent step size as

$$\rho_i = \arg\min_{\rho_i} \mathcal{L}\big(\hat{f}_i(\mathbf{x}) + \rho_i g_i(\mathbf{x}^{(i)}), \mathbf{w}\big) \tag{4.8}$$

    Update the estimate of $\hat{f}_i(\mathbf{x}^{(i)})$ as

$$\hat{f}_i(\mathbf{x}^{(i)}) \leftarrow \hat{f}_i(\mathbf{x}^{(i)}) + \rho_i g_i(\mathbf{x}^{(i)}) \tag{4.9}$$

  end

  Update $\mathbf{w}$ as Eq. 4.10 and Eq. 4.11.

end

$\hat{f}(\mathbf{x}) = \mathbf{P}(\sum \omega_i \hat{f}_i(\mathbf{x}^{(i)}))$

Algorithm 1: Gradient Boosting Consensus

where $\mathbf{L}\big(f_i(\mathbf{x}_a), f_i(\mathbf{x}_b)\big)$ is the pairwise loss that evaluates the difference between the two predictions $f_i(\mathbf{x}_a)$ and $f_i(\mathbf{x}_b)$. The idea behind Eq. 4.11 is to evaluate whether a graph can link similar instances together. If most of the connected instances have similar predictions, the graph is considered to be informative. Note that both the weights in Eq. 4.10 and the weights in Eq. 4.11 are updated at each iteration. By replacing them into Eq. 4.4, one can observe that the objective function of the GBC model is adaptively updated at each iteration. In other words, at the initial step, each data source will be given equal weights; but after several iterations, informative data sources will have higher learning weights, and the objective function will "trust" more the informative data sources. Note that this is a very important setting in dealing with the case that the consensus assumption does not hold. In this situation, the weight learning process will drop the weights of other data sources close to zero. The whole model will be degraded to a normal GBDT.

4.4 Generalization bounds In this section, we consider the incompatibility framework in [34] and [35] to explain the proposed GBC model. Specifically, we show that the weight learning process described in Section 4.3 can help reduce an error bound. For the sake of simplicity, we consider the case where we have two data sources $\mathcal{X}_1$ and $\mathcal{X}_2$, and the case with more data sources can be analyzed with similar logic. Note that the goal is to learn a pair of predictors $(f_1; f_2)$, where $f_1 : \mathcal{X}_1 \to \hat{\mathcal{Y}}$ and $f_2 : \mathcal{X}_2 \to \hat{\mathcal{Y}}$, and $\hat{\mathcal{Y}}$ is the prediction space. Further denote $\mathcal{F}_1$ and $\mathcal{F}_2$ as the hypothesis classes of interest, consisting of functions from $\mathcal{X}_1$ (and, respectively, $\mathcal{X}_2$) to the prediction space $\hat{\mathcal{Y}}$. Denote by $L(f_1)$ the expected loss of f_1, and $L(f_2)$ is similarly defined. Let a Bayes optimal predictor with respect to loss L be denoted as f^*. We now apply the incompatibility framework for the multi-view setting [34] to study GBC. We first define the incompatibility function $\chi : \mathcal{F}_1 \times \mathcal{F}_2 \to \mathbb{R}^+$, and some $t \geq 0$ as those pairs of functions which are compatible to the tune of t, which can be written as:

$$\mathcal{C}^\chi(t) = \{(f_1, f_2) : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2 \text{ and } \mathbb{E}[\chi(f_1, f_2)] \leq t\}$$

Intuitively, the function $\mathcal{C}^\chi(t)$ captures the set of function pairs f_1 and f_2 that are compatible with respect to a "maximal expected difference" t. From [34], it is proven that there exists a symmetric function $d : \mathcal{F}_1 \times \mathcal{F}_2$, and a monotonically increasing non-negative function Φ on the reals such that for all f,

$$\mathbb{E}[d(f_1(x); f_2(x))] \leq \Phi(L(f_1) - L(f_2))$$

With these functions at hand, we can derive the following theorems:

THEOREM 4.1. *Let $|L(f_1) - L(f^*)| < \epsilon_1$ and $|L(f_2) - L(f^*)| < \epsilon_2$, then for the incompatibility function $\mathcal{C}^\chi(t)$, if we set $\chi = d$, for $t = c_d(\Phi(\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}) + \Phi(\epsilon_{bayes}))$ where c_d is a constant depends on the function d [34], we have*

$$\inf_{(f_1,f_2)\in\mathcal{C}^\chi(t)} L_{GBC}(f_1, f_2) \leq L(f^*) + \epsilon_{bayes} + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1 + \epsilon_2}} \tag{4.12}$$

Proof. Note that $|L(f_1) - L(f^*)| < \epsilon_1$ and $|L(f_2) - L(f^*)| < \epsilon_2$, and the proposed model GBC adopts a weighted strategy linear to the expected loss, which is approximately $L_{\text{GBC}}(f_1, f_2) = \frac{\epsilon_2}{\epsilon_1+\epsilon_2}L(f_1) + \frac{\epsilon_1}{\epsilon_1+\epsilon_2}L(f_2)$. According to Lemma 8 in [35], we have $E[\chi(f_1, f_2)] \leq c_d^2(\Phi(\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}) + \Phi(\epsilon_{\text{bayes}}))$, and

$$\min_{(f_1,f_2)\in\mathcal{C}^\chi(t)} L_{\text{GBC}}(f_1, f_2) \leq L_{\text{GBC}}(f*_1, f*_2) + \epsilon_{\text{bayes}} \tag{4.13}$$

With Lemma 7 in [35], we can get

$$\min_{(f_1,f_2)\in\mathcal{C}^\chi(t)} L_{\text{GBC}}(f_1, f_2) \leq L(f^*) + \epsilon_{\text{bayes}} + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1 + \epsilon_2}} \tag{4.14}$$

□

Similarily, we can derive the error bound of GBC in a transductive setting.

THEOREM 4.2. *Consider the transductive formula Eq. 4 in [35]. Given the regularized parameter $\lambda > 0$, we denote $L^\lambda(f)$ as the expected loss with the regularized parameter λ. If we set $\lambda_c = \frac{\lambda}{4(K+\lambda)^2\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}}$ then for the pair of functions $(f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$ returned by the transductive learning algorithm, with probability at least $1-\delta$ over labeled samples,*

$$\begin{aligned} L^\lambda_{GBC}(f_1, f_2) &\leq L^\lambda(f^*) + \frac{1}{\sqrt{n}}\Big(2 + 3\sqrt{\frac{ln(\frac{2}{\delta})}{2}}\Big) \\ &+2C_{Lip}\hat{R}(\hat{\mathcal{C}}^\chi(\frac{1}{\lambda_c})) + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \end{aligned} \tag{4.15}$$

where n is the number of labeled examples, and C_{Lip} is the Lipschitz constant for the loss, and $\hat{R}(\hat{\mathcal{C}}^\chi(\frac{1}{\lambda_c}))$ is a term bounded by the number of unlabeled examples and the bound of the losses.

Proof. We first note that $|L^\lambda(f_1) - L^\lambda(f^*)| < \epsilon_1$ and $|L(f_2)^\lambda - L^\lambda(f^*)| < \epsilon_2$. Similar to the logic in Theorem 4.1, GBC employs a weighting strategy which is linear to the expected loss: $L^\lambda_{\text{GBC}}(f_1, f_2) = \frac{\epsilon_2}{\epsilon_1+\epsilon_2}L^\lambda(f_1) + \frac{\epsilon_1}{\epsilon_1+\epsilon_2}L^\lambda(f_2)$. With Lemma 7 in [35], we then have

$$\begin{aligned} &|L^\lambda(f_1) - L^\lambda_{\text{GBC}}| + |L^\lambda(f_1) - L^\lambda_{\text{GBC}}| \\ &\leq \sqrt{\frac{\epsilon_2}{\epsilon_1+\epsilon_2}}\sqrt{\epsilon_1} + \sqrt{\frac{\epsilon_1}{\epsilon_1+\epsilon_2}}\sqrt{\epsilon_2} \\ &= 2\sqrt{\frac{\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \end{aligned} \tag{4.16}$$

Hence, we can further get the following relationship:

$$\begin{aligned} E[\chi(f_1, f_2)] &\leq c_d^2(E[\chi(f_1, y_1)] + E[\chi(y_1, y_2)] + E[\chi(f_2, y_2)]) \\ &\leq c_d^2(L^\lambda(f_1) - L^\lambda(f^*) + L^\lambda(f_2) - L^\lambda(f^*) + 2\Phi(\epsilon_{\text{bayes}})) \\ &\leq c_d^2(\Phi(\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}) + \Phi(\epsilon_{\text{bayes}})) \end{aligned} \tag{4.17}$$

and

$$\min_{(f_1,f_2)\in\mathcal{C}^\chi(t)} L^\lambda_{\text{GBC}}(f_1, f_2) \leq L^\lambda_{\text{GBC}}(f*_1, f*_2) + \epsilon_{\text{bayes}} \tag{4.18}$$

We then have

$$\begin{aligned} L^\lambda_{\text{GBC}}(f_1, f_2) &\leq L^\lambda(f^*) + \frac{1}{\sqrt{n}}\Big(2 + 3\sqrt{\frac{ln(\frac{2}{\delta})}{2}}\Big) \\ &+2C_{Lip}\hat{R}(\hat{\mathcal{C}}^\chi(\frac{1}{\lambda_c})) + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \end{aligned} \tag{4.19}$$

□

Note that Theorem 4.1 and Theorem 4.2 derive the error bounds of GBC in inductive and transductive setting respectively. In effect, the weighting strategy reduces the last term of

the error bound to $\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}$, as compared to the equal-weighting strategy whose last term is $\sqrt{\frac{\epsilon_1+\epsilon_2}{2}}$ [34]. Hence, the weighting strategy induces a tighter bound since $\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \leq \sqrt{\frac{\epsilon_1+\epsilon_2}{2}}$. It is important to note that if the the predictions of different data sources vary significantly ($|\epsilon_1 - \epsilon_2|$ is large), the proposed weighting strategy has a much tighter bound than the equal-weighting strategy. In other words, if there are some noisy data sources that potentially lead to large error rate, GBC can effectively reduce their effect. This is an important property of GBC to handle noisy data sources. This strategy is evaluated empirically in the next section.

5 Experiments

In this section, we report four sets of experiments that were conducted in order to evaluate the proposed GBC model applied to multiple data sources. We aim to answer the following questions:

- Can GBC make good use of multiple data sources? Can it beat other more straightforward strategies?
- What is the performance of GBC if there exist non-overlapping instances in different data sources?
- How does GBC perform on big dataset?

5.1 Datasets The aim of the first set of experiments is to predict movie ratings from the IMDB database.[2] Note that there are 10 data sources in this task. For example, there is a data source about the plots of the movies, and a data source about the techniques used in the movies (*e.g.*, 3D IMAX). Furthermore, there are several data sources providing different graph relational data about the movies. For example, in a director graph, two movies are connected if they have the same director. A summary of the different data sources can be found in Table 2. It is important to note that each of the data sources may provide certain useful information for predicting the ratings of the movies. For instance, the Genre database may reflect that certain types of movies are likely to have high ratings (*e.g.*, Fantasy); the Director graph database implicitly infers movie ratings from similar movies of the same director (*e.g.*, Steven Spielberg has many high-rating movies.). Thus, it is desirable to incorporate different types of data sources to give a more accurate movie rating prediction. This is an essential task for online TV/movie recommendation, such as the famous $1,000,000 Netflix prize [36].

The second set of experiments is about handwritten number recognition. The dataset contains 2000 handwritten numerals ("0"–"9") extracted from a collection of Dutch utility maps.[3] The handwritten numbers are scanned and digitized as binary images. They are represented in terms of the following seven data sources with different vector-based feature spaces: (1) 76 Fourier coefficients of the character shapes, (2) 216 profile correlations, (3) 64 Karhunen-Love coefficients, (4) 240 pixel averages in 2×3 windows, (5) 47 Zernike moments, (6) a graph dataset constructed from the morphological similarity (*i.e.*, two objects are connected if they have similar morphology appearance), and (7) a graph generated with the same method as (6), but with random Gaussian noise imposed in the morphological similarity. This dataset is included to test the performance

[2] http://www.imdb.com/

[3] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

Table 2: IMDB Movie Rating Prediction

Data source	Type of features
Quote Database	Text
Plot Database	Text
Technology Database	Nominal
Sound Technology Database	Nominal
Running Time Database	Real
Genre Database	Binary
Actor Graph	Graph
Actress Graph	Graph
Director Graph	Graph
Writter Graph	Graph

of GBC on noisy data. The aim is to classify a given object to one of the ten classes ("0"–"9"). The statistics of the dataset are summarized in Table 3.

The third set of datasets is downloaded from the UMD collective classification database [4]. The database consists of 1293 different attacks in one of the six labels indicating the type of the attack: arson, bombing, kidnapping, NBCR attack, weapon attack and other attack. Each attack is described by a binary value vector of attributes whose entries indicate the absence or presence of a feature. There are a total of 106 distinct vector-based features, along with three sets of relational features. One set connects the attacks together if they happened in the same location; the other connects the attacks if they are planned by the same organization. In order to perform robust evaluation of the proposed GBC model, we add another data source based on the vector-based dataset, but with a random Gaussian noise $\mathcal{N}(0, 1)$. Again, this is to test the capability of the proposed model to handle noise.

The fourth set of datasets is collected from a tier-1 telegraph network provider in the U.S. We study a subset of the demographic database with over 500,000 anonymous users. The database records 196 demographic features per user, which includes education level, age group, language, hobbies, *etc.*. As the objective in [37], we aim at predicting four demographic features, which include the age group, gender, credit level, and whether the user rents the house. For the task of predicting the age group and credit level, we only concern whether the user belongs to a specific group of interest (*e.g.*, mid age and good credit level). As a result, all four tasks have binary classification labels. Furthermore, we construct 90 different social graphs from the phone call networks similar to the ones introduced in [38]. These generated social graphs are considered to be close approximations to the real-world social connections, and they cover the social connections among anonymous users in different time periods. In summary, we have one vector-based demographic dataset and 90 graph datasets, and each of them involves a subset of the 500,000 anonymous users. Another very important characteristic is that the dataset contains substantial missing values, owing to the difficulty of obtaining the demographic features (*e.g.*, love fishing? speak Japanese?, *etc.*). In the dataset, over 50% of samples contain over 60% of missing values. Classic feature based algorithms such as SVM cannot easily handle this case. On the

[4] http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html

contrary, GBC is designed to fit to the situation with missing values and with many heterogeneous data sources. We design four prediction tasks to predict four important demographic features of the samples, and they include the prediction of the age group, gender, credit level, and whether the user rents the house. Our objective is then to evaluate the four prediction tasks, all of which involve big and heterogeneous data.

5.2 Comparison Methods and Evaluations It is important to emphasize again that there is no previous model that can handle the same problem directly; i.e., building a learning model from multiple graphs and multiple vector-based datasets with some non-overlapping instances. Furthermore, as far as we know, there is no state-of-the-art approaches that use the benchmark datasets described in the previous section in the same way. For instance, in the movie prediction dataset, we crawl the 10 data sources directly from IMDB and use them collectively in learning. In the case of the number recognition dataset, we have two graph data sources, which are different from previous approaches that only look at vector-based features [39], clustering [40], or feature selection problems [41]. In order to evaluate the proposed GBC model, we design a straightforward comparison strategy, which is to directly join all features together. In other words, given the sources with vector-based features $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(p)}$ and the adjacency matrices of the graphs $\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(q)}$, the joined features can be represented as follows:

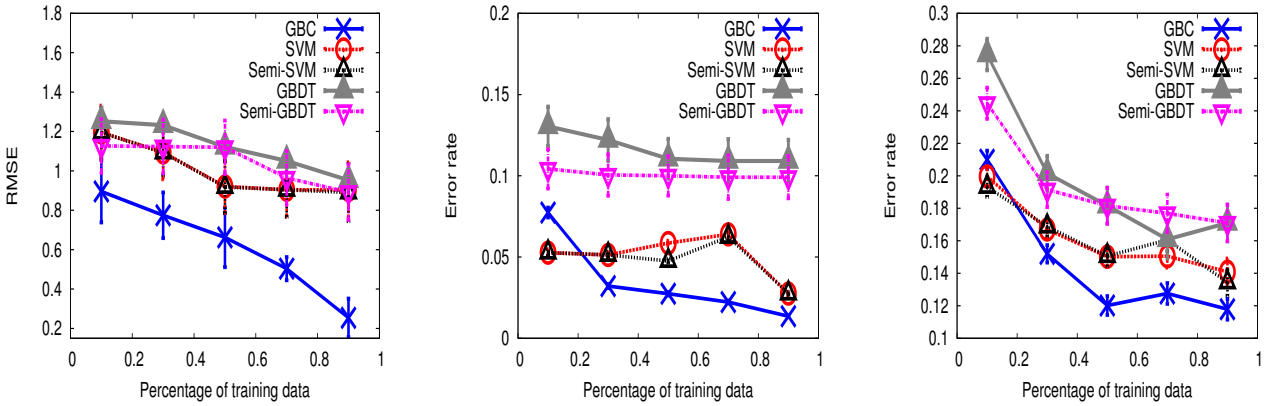
$$\mathbf{X} = [\mathbf{X}^{(1)^T}, \cdots, \mathbf{X}^{(p)^T}, \mathbf{M}^{(1)^T}, \cdots, \mathbf{M}^{(q)^T}]^T \tag{5.20}$$

Since there is only one set of joined features, traditional learning algorithms can be applied on it to give predictions (each row is an instance; each column is a feature from a specific source). We include support vector machines (SVM) in the experiments as it is used widely in practice. Note that in GBC, the consensus term in Eq. 4.2 and the graph similarity term in Eq. 4.3 can use unlabeled data to improve the learning. Hence, we also compare it with semi-supervised learning models. Specifically, semi-supervised SVM (Semi-SVM) with a self-learning technique [42] is used as the second comparison model. Note that we have four tasks in the experiment where one of them (*i.e.*, the movie rating prediction task) is a regression task. In this task, regression SVM [43] is used to give predictions. Additionaly, since the proposed model is derived from gradient boosting decision trees, GBDT [26] is used as the third comparison model, and its semi-supervised version [42] is included as well. It is important to note that in order to use the joined features from Eq. 5.20, these comparison models require that there is no non-overlapping instances. In other words, all data sources should have records of all instances; otherwise, the joined features will have many missing values since some data sources may not have records of the corresponding instances. To evaluate GBC more comprehensively, we thus conducted the experiments on two settings:

- Uniform setting: the first setting is to force all data sources to contain records of all instances. We only look at the instances that have records in all data sources. Table 3 presents the statistics of the datasets in this setting. In this case, we can easily join the features from different sources as in Eq. 5.20. Note that we do not perform the uniform setting in AT&T's big dataset. The reason is that there is only a couple of users that appear in all 91 different data sources. The uniform setting thus cannot generate statistically significant results on AT&T's big dataset.

Table 3: Data Descriptions

Task	# of data	# of data sources	Average dimension	Predictions
Movie Rating Prediction	3000	10 (4 graph and 6 others)	896 (×10)	Regression
Number Recognition	2000	7 (2 graph and 5 others)	378 (×7)	10 labels
Terrorist Attack Classification	1293	4 (2 graph and 2 others)	422 (×4)	6 labels
AT&T Datasets	500,000	91	499,997 (×91)	2 labels



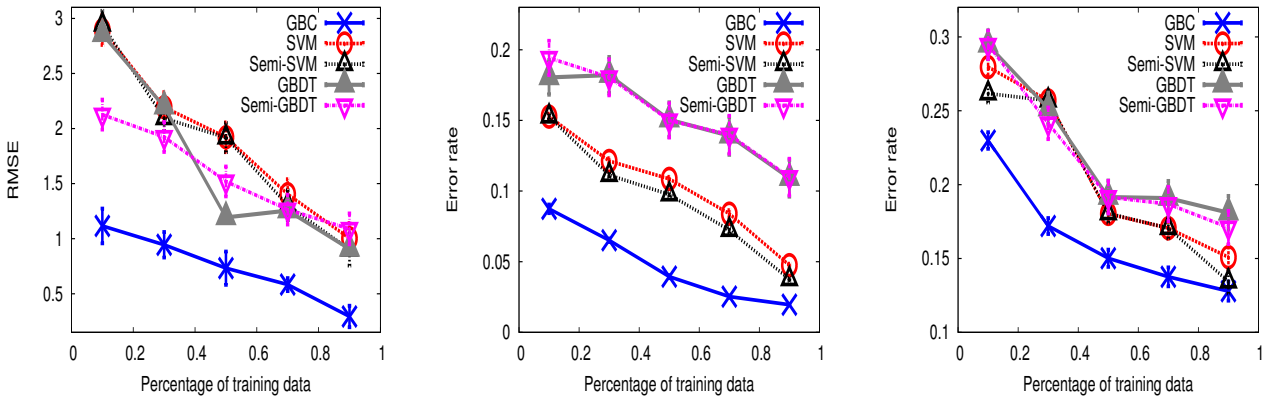
(a) Movie rating prediction (b) Number recognition (c) Terrorist attack detection

Figure 3: All data sources record the same set of objects (overlapping objects with different features).

- Non-overlapping setting: the second setting is to allow different data sources to have some non-overlapping instances. Thus, an instance described in one data source may not appear in other data sources. This setting is more realistic, as the example in Fig. 1. The proposed GBC model is able to handle this case, since it allows non-overlapping instances. However, for the comparison method, there will be many missing values in the joined features as discussed above. In this case, we replaced the missing values with the average values of the corresponding features. In this setting, 30% of the instances do not have records in half of the data sources.

We conducted experiments on the above two settings. During each run, we randomly selected a certain portion of examples as training data, keeping the others as test data. For the same training set size, we randomly selected the set of training data 10 times and the rests were used as test data, and the results were averaged over the 10 runs. The experiment results are reported with different training set sizes. Note that the proposed GBC model can be used for both classification and regression. We used error rate to evaluate the results for classification tasks, and root mean square error (RMSE) for regression tasks.

5.3 Analysis of the Experiments Our aim is to study the performance of the proposed GBC model in the two setting described above: uniform and non-overlapping settings. The experiment results are summarized in Fig. 3 and Fig. 4, respectively. The x-axes record different percentage of training data (while the remainder of the data is used for evaluation), and the y-axes report



(a) Movie rating prediction (b) Number recognition (c) Terrorist attack detection

Figure 4: Each data source is independent (with 30% of overall non-overlapping instances).

the errors of the corresponding learning model.

We observe two major phenomena in the experiments. Firstly, the proposed GBC model effectively reduces the error rate as compared to the other learning models in both settings. It is especially obvious in the movie rating prediction dataset where 10 data sources are used to build the model. In this dataset, GBC reduces the error rate by as much as 80% in the first setting (when there are 90% of training instances), and 60% in the second setting (when there are 10% of training instances). This shows that GBC is especially advantageous when a large number of data sources are available. We further analyze this phenomenon in the next section with Table 4. On the other hand, the comparison models have to deal with a longer and noisier feature vector. GBC beats the four approaches by judiciously reducing the noise (as discussed in Section 4.3 and 4.4). Secondly, we can observe that GBC outperforms the other approaches significantly and substantially in the second setting (Fig. 4) where some instances do not have records on all data sources. As analyzed in the previous section, this is one of the advantages of GBC over the comparison models that have to deal with missing values.

It is also interesting to analyze the performance of GBC on the demographic prediction task. Recall that the learning task has at least three challenges. First, it contains a large set of samples (over 500,000) as compared to normal machine learning tasks. Second, it has 91 different data sources, among which one is a demographic dataset with around 200 features, and the other 90 data sources are social graphs. As a result, it generates over 45,000,000 joined features (as in Eq. 5.20). Third, it contains substantial missing values (50% of samples miss 60% of features), which brings in more difficulty in finding the valuable information. This is also the reason we cannot conduct uniform setting on this big dataset, since there is only a couple of users that appear in all 91 different data sources. The uniform setting thus cannot generate statistically significant results. All the experiments reported in Fig. 6 were run on a distributed computing system with the Condor framework. It can be clearly observed that GBC outperforms the other comparison models in all four tasks. For instance, in the task of predicting genders, the error rate has been reduced around 20% when there is 50% of training data, and 25% when there is 90% of training examples. The good performance of GBC comes from its unique design to

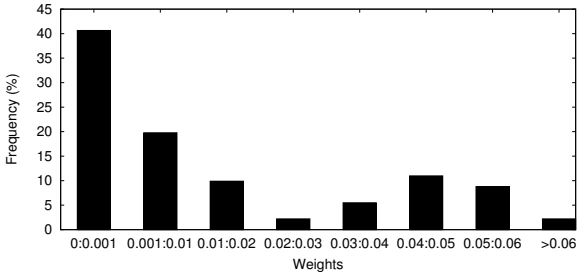


Figure 5: Frequency of the Learned Weights

distill useful information from multiple heterogeneous data sources. Each of the individual data source may be very noisy owing to the missing values and unknown data quality. However, GBC is able to distinguish the useful data sources for the learning tasks, and judiciously uses them in improving learning results. The distribution of the learned weights of the data sources are plotted in Fig. 5. It can be shown that among the 91 data sources, some of them are very useful (with large weights), while many of them are quite noisy and are assigned low weights. GBC is able to judiciously treat them differently to achieve a better model.

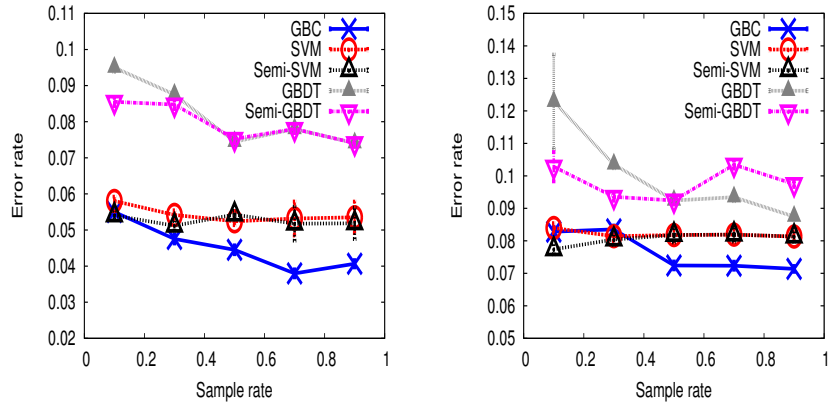
6 Discussion

In this section, we would like to answer the following questions:

- To what extent GBC helps to integrate the knowledge from multiple sources, compared to learn from each source independently? Specifically, how do the principles of consensus and connectivity similarity help GBC?
- Is the weight learning algorithm necessary?
- Do we need multiple data sources? Does the number of data sources affects the performance?

In GBC, both λ_0 and λ_1 (from Eq. 4.4) are determined by cross-validation. In the following set of experiments, we tune the values of λ_0 and λ_1 in order to study the specific effects of the consensus and connectivity terms. We compare the proposed GBC model with three algorithms on the number recognition dataset in the uniform setting (*i.e.*, all data sources contain all instances):

- The first comparison model is to set λ_0 to zero, and determine λ_1 by cross-validation. In this case, the consensus term is removed. In other words, the algorithm can only apply the connectivity similarity principle. We denote this model as "GBC without consensus".
- The second comparison model is to set λ_1 to zero, and let λ_0 be determined by cross-validation. In other words, the connectivity similarity term is removed, and the algorithm only depends on the empirical loss and the consensus term. We denote this model as "GBC without graph".
- The third comparison model is to set both λ_0 and λ_1 to 0. Hence, the remaining term is the empirical loss, and it is identical to the traditional GBDT model.



(a) Credit Level (b) Age Group

(c) Gender (d) Renter

Figure 6: Demographic Prediction

The empirical results on the number recognition task are presented in Fig. 7. It can be observed that all three models outperforms traditional GBDT. We draw two conclusions from this experiment. First, as was already observed in previous experiments, learning from multiple sources is advantageous. Specifically, the GBDT model builds classifiers for each of the data source independently, and average the predictions at the last step. However, it does not "communicate" the prediction models during training. As a result, it has the worst performance in Fig. 7. Second, both the principle of consensus and the principle of connectivity similarity improve the performance. Furthermore, it shows that the connectivity similarity term helps improve the performance more when the number of training data is limited. For example, when there are only 10% of training instances, the error rate of GBC with only the connectivity term (*i.e.*, GBC without consensus) is less than 9%, while that of GBC with only the consensus term (*i.e.*, GBC without Graph) is around 13%. This is because when the number of labeled training data is limited, the graph connectivity serves as a more important source of information when connecting unlabeled data with the limited labeled data. Again, it is important to note that in GBC, the weights of different data sources are adjusted at each iteration. The aim is to assign higher weights to the data sources that contain useful information, and filter out the noisy sources. This step is analyzed as an important one in Theorem 4.1 since it can help reduce the

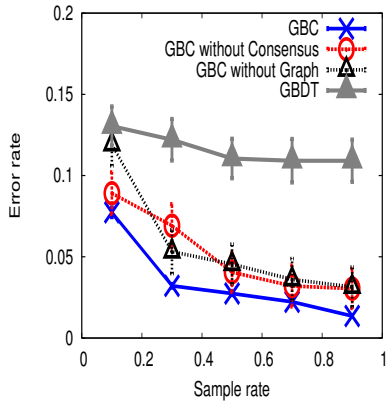


Figure 7: How do consensus principle and connectivity similarity principle help?

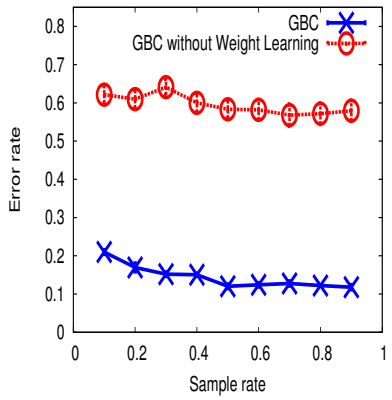


Figure 8: Why is the weighting strategy necessary?

upper bound of the error rate. We specifically evaluate this strategy on the terrorist detection task. Note that in order to perform a robust test, one of the vector-based sources contains Gaussian noise, as described in Section 5.1. The empirical results on the terrorist detection task are presented in Fig. 8. It can be clearly observed that the weighting strategy is reducing the error rate by as much as 70%. Hence, an appropriate weighting strategy is an important step when dealing with multiple data sources with unknown noise.

It is also interesting to evaluate to what extent GBC performance is improved as the number of data sources increases. For this purpose, the movie rating prediction dataset is used as an example. We first study the case when there is only one data source. In order to do so, we run GBC on each of the data source independently, then on 2 and 4 data sources. In the experiments with 2 data sources, we randomly selected 2 sources from the pool (Table 2) as inputs to GBC. These random selections of data sources were performed 10 times, and the average error is reported in Table 4. A similar strategy was implemented to conduct the experiment with 4 data sources. In Table 4, the results are reported with different percentages of training data, and the best performances are highlighted with bold letters. It can be observed that the performance with only one data source is the worst, and has high root mean square error and high variance. With more data sources available, the performance of GBC tends to be better. This is because

Table 4: Effect of different number of sources. Reported results are RMSE with variance in parenthesis.

# of sources	10%	30%	50%	70%	90%
1	1.356 (0.358)	1.223 (0.290)	1.192 (0.257)	1.189 (0.258)	1.077 (0.223)
2	1.255 (0.237)	1.247 (0.293)	1.193 (0.229)	1.193 (0.150)	0.973 (0.216)
4	1.135 (0.138)	**0.914 (0.114)**	**0.732 (0.153)**	0.593 (0.105)	0.314 (0.092)
all	**1.115 (0.158)**	0.945 (0.116)	**0.732 (0.152)**	**0.583 (0.059)**	**0.294 (0. 097)**

each data source provides complementary information useful to build a comprehensive model of the whole dataset.

7 Conclusion

This paper studies the problem of building a learning model from heterogeneous data sources. Each source can contain traditional vector-based features or graph relational features, with potentially non-overlapping sets of instances. As far as we know, there is no previous model that can be directly applied to solve this problem. We propose a general framework derived from gradient boosting, called gradient boosting consensus (GBC). The basic idea is to solve an optimization problem that (1) minimizes the empirical loss, (2) encourages the predictions from different data sources to be similar, and (3) encourages the predictions of connected data to be similar. The objective function is solved by stochastic gradient boosting, with an incorporated weighting strategy to adjust the importance of different data sources according to their usefulness. Four sets of experiments were conducted, including movie rating prediction, number recognition, terrorist detection, and AT&T's challenging tasks with over 500,000 samples and 91 data sources. We show that the proposed GBC model substantially reduce prediction error rate by as much as 80%. Finally, several extended experiments are conducted to study specific properties of the proposed algorithm and its robustness.

8 Acknowledgements

Part of the work was done when Xiaoxiao Shi was a summer intern at AT&T Labs. This work is also supported in part by NSF through grants IIS-0905215, CNS-1115234, IIS-0914934, DBI-0960443, and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, Google Mobile 2014 Program and KAU grant.

References

[1] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *AAAI/IAAI*, pp. 187–192, 2002.

[2] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, pp. 92–100, 1998.

[3] S. Oba, M. Kawanabe, K. Müller, and S. Ishii, "Heterogeneous component analysis," in *NIPS*, 2007.

[4] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *CIKM*, pp. 86–93, 2000.

[5] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in *SDM*, pp. 822–833, 2008.
[6] J. Gao, W. Fan, Y. Sun, and J. Han, "Heterogeneous source consensus learning via decision propagation and negotiation," in *KDD*, pp. 339–348, 2009.
[7] D. Agarwal, B. Chen, and B. Long, "Localized factor models for multi-context recommendation," in *KDD*, pp. 609–617, 2011.
[8] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.
[9] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data.," in *Proceedings of the International Conference on Machine Learning*, 2001.
[10] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data.," in *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 2002.
[11] H. Eldardiry and J. Neville, "Across-model collective ensemble classification," in *AAAI*, 2011.
[12] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *ICML*, pp. 81–88, 2007.
[13] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
[14] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *KDD*, pp. 283–291, 2008.
[15] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *ICDM*, pp. 1049–1054, 2010.
[16] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *NIPS*, pp. 137–144, 2006.
[17] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *ACL*, 2007.
[18] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *KDD*, pp. 210–219, 2007.
[19] R. K. Ando and T. Zhang, "A high-performance semi-supervised learning method for text chunking," in *ACL*, 2005.
[20] J. Blitzer, R. T. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *EMNLP*, pp. 120–128, 2006.
[21] H. D. III, "Frustratingly easy domain adaptation," *CoRR*, vol. abs/0907.1815, 2009.
[22] S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *ICML*, pp. 489–496, 2007.
[23] T. Jebara, "Multi-task feature and kernel selection for svms," in *ICML*, 2004.
[24] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *ICML*, pp. 1120–1127, 2008.
[25] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.
[26] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.
[27] T.-Y. Liu, *Learning to Rank for Information Retrieval*. Springer, 2011.
[28] Y. Koren, "The bellkor solution to the netflix grand prize," 2009.
[29] J. Gao, W. Fan, Y. Sun, and J. Han, "Heterogeneous source consensus learning via decision propagation and negotiation," in *KDD*, pp. 339–347, 2009.
[30] X. Shi and P. S. Yu, "Dimensionality reduction on heterogeneous feature space," in *ICDM*, 2012.
[31] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning*

Research, vol. 12, pp. 2211–2268, 2011.
[32] D. P. Bertsekas, *Nonlinear Programming (Second ed.)*. Cambridge, MA.: Athena Scientific, 1999.
[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
[34] M. Balcan and A. Blum, "A pac-style model for learning from labeled and unlabeled data," in *COLT*, pp. 111–126, 2005.
[35] K. Sridharan and S. M. Kakade, "An information theoretic framework for multi-view learning," in *COLT*, pp. 403–414, 2008.
[36] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[37] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen, "Demographic prediction based on user's browsing behavior," in *WWW*, pp. 151–160, 2007.
[38] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "A first look at cellular machine-to-machine traffic: large scale measurement and characterization," in *SIGMETRICS*, pp. 65–76, 2012.
[39] M. van Breukelen and R. Duin, "Neural network initialization by combined classifiers," in *ICPR*, pp. 16–20, 1998.
[40] X. Z. Fern and C. Brodley, "Cluster ensembles for high dimensional clustering: An empirical study," *Journal of Machine Learning Research.*, vol. 22, no. 8, pp. 888–905, 2004.
[41] X. He and P. Niyogi, "Locality preserving projections," in *NIPS*, 2003.
[42] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2009.
[43] P. Laskov, "An improved decomposition algorithm for regression support vector machines," in *NIPS*, pp. 484–490, 1999.