Robust Prediction from Multiple Heterogeneous Data Sources with Partial Information

Mohammad S. Aziz
Wayne State University
Detroit, MI-48201, USA
maziz@wayne.edu

Chandan K. Reddy
Wayne State University
Detroit, MI-48201, USA
reddy@cs.wayne.edu

ABSTRACT

Significant research efforts for robust integration of information from multiple sources are being pursued at a rapid pace. However, the information in heterogeneous sources is often incomplete and hence making the maximum use of all the available information is a challenging problem. Most of the recent research on data integration have been primarily focused on the cases where the information is available across all the different sources and can not effectively integrate sources in the presence of partial information. We develop an ensemble method that boosts the decisions made from different models on individual sources and obtain robust results for the task of class prediction. We propose a heterogeneous boosting framework that uses all the available information even if some of the sources do not provide any information about some objects. We demonstrate the effectiveness of the proposed framework for the problem of gene function prediction and compare to the state-of-the-art methods using several real-world biological datasets. We also show that the proposed method outperforms any kind of imputation schemes that are widely used while integrating data with partial information.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Measurement, Design

Keywords

Classification, boosting, data integration, gene function prediction.

1. INTRODUCTION

Integrating information from multiple sources and making combined decisions from these sources is becoming a common task across several disciplines. Although there are several works proposed for heterogeneous data integration, there is no systematic approach that enhances the prediction performance for different applications. Different ensemble strategies [3] for the integration such as decision templates, weighted majority voting, bagging, boosting, and random forests have been extensively used for integrating different models. Most of these works assume that the information about a data object is available from each of the model. In addition, these methods typically assume that the different models are built from the same type of feature sets (or data sources).

In practice, multiple heterogeneous sources do not contain all the required information about a particular data object. Some sources deliver information about some of the data objects whose information is not available from other sources. In other words, when all the sources are combined, there will be some missing information about certain data objects with respect to some sources. Most of the current research work in data integration primarily focuses on integrating information when all the sources contain the information about a data object. That is, for the sake of convenience, researchers primarily deal with common objects that are available in all the data sources. Considering only the common information will potentially harm the class prediction when there are several data objects with partial information. Some work on handling partial information is available in the kernel methods literature [11] where the kernel matrix is integrated to combine the information from multiple sources. Most of these methods treat it as a missing data problem to calculate the missing features with the help of observed features to subsequently compute the kernel matrix.

In this paper, we primarily focus on improving the performance on functional classification task by utilizing multiple sources of information about a set of genes. In the field of Functional Genomics, the functional classification of unannotated genes and subsequently, the improvement of the existing gene functional annotation catalogs is an important and challenging problem [12]. Functional classification plays a vital role in molecular biology due to its ability to detect previously unknown role of genes and their products in physiological and pathological processes. Different types of biomolecular data, ranging from expression profiles to phylogenetic gene-specific evolution rates and many others are available to classify gene functions. Such vast amounts of data, in principle, can provide useful information for the automated assessment of the functional role of genes. The extent to which the classification performance can be improved significantly depends on specific type of experimental data

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

and varies for specific gene and the particular bio-molecular process under investigation. One of the key challenges in this domain is that many sources contain only partial information and one can rarely see all the information available in a given source. Hence, it is critical in this domain to develop methods that work only with such partial information.

2. RELATED BACKGROUND

2.1 Notations Used

Table 1: Notations used in this paper.

Notation	Description
N	Number of datasets
m	Number of total data objects
D_i	i^{th} dataset
m_i	Number of data objects in D_i
d_k	k^{th} data object
w_{ki}	Weight of the data object d_k for D_i
C_{ij}	Weak classifier at j^{th} iteration for D_i
c_{ij}	Weight of the weak classifier C_{ij}

2.2 Methods for Data Integration

Several approaches for heterogeneous data integration have been proposed in the literature. One of the earlier approaches was to integrate multiple sources into one combined dataset and perform the classification task on it. Vector Space Integration (VSI) [2] is one of the popular techniques that fall into this category. Other such methods were based on modeling networks of functional relationships between proteins where graphical models provide a probabilistic framework for data integration [5]. These methods are often referred as *early integration*. In general, these methods do not make use of any class information during the integration and hence, do not yield better classification performance. Due to different properties and heterogeneity of the features, it is not always possible to build graphical models or concatenate vector spaces. *Intermediate integration* is typically based on kernel fusion (KF) methods. The integration is performed during the training phase itself. Individual kernels on different sources are learned and the final classifier is built on the composite kernel which is obtained after combining the individual ones [7]. To exploit the heterogeneity of the data, often weighted functional linkage graph is generated by different sources of information [13]. Another family of successful approaches often referred as *late integration*, which typically model individual sources and then combine the knowledge from these individual models and builds a final classifier [9, 6]. Methods such as decision templates, different types of weighted majority voting using linear or logarithmic weight combination, and ensemble methods like bagging, boosting and random forests fall into this category [8]. The intuition for using the ensemble technique is that, when the base classifiers used in the ensemble are diverse, they are expected to make different errors. Hence, the ensemble output produced by these classifiers is expected to reduce the overall misclassifications.

2.3 Methods for Data Imputation

Some of the popular methods used for data imputation in the context of integrating multiple sources using kernel fusion techniques are as follows: *(i) Unconditional Mean Imputation (KF_UMI):* For a data object that is not in a particular source, the feature values are imputed with the average of the feature values of the objects that are present in that source. After getting all the feature values, the kernel matrix is aggregated to get a single matrix and and a SVM classifier if built on this matrix to make the final prediction. $f^j_{d_k} = \frac{1}{|S|}\sum_{d\in S} f^j_d$, where, S is the set of data that has the value for j^{th} feature. *(ii) Weighted Summation Imputation (KF_WSI):* The feature values for the data are not imputed in the source. Rather, for a data object that is missing in a source, the kernel matrix entries of that object for this kernel are imputed as a weighted combination of the average of the entries of that particular kernel matrix and the average of the entries for that objects in other kernel matrices where the data object is present. In our experiment, we used 50% weight for both these values. *(iii) Nearest Neighbor Imputation (KF_NNI):* In this method, the kernel matrix entries are directly imputed rather than imputing the feature values. First, the kernel matrices for different sources are generated from the data objects present in those sources. At this juncture, for a data object that is not present in a particular source will not have any kernel matrix entry. Using the other sources where the feature values are present for that data object, the nearest neighbor is obtained. Then, in the source where that data object was missing, the kernel matrix entry of the nearest neighbor is replicated.

3. OUR PROPOSED ALGORITHM

To ensure the improved accuracy when combining models from multiple sources, we propose to boost the decisions from these individual sources using a modification to AdaBoost [4] algorithm. AdaBoost is an efficient, simple and easy to manipulate additive modeling technique that can potentially use any available weak learner. Boosting algorithms combine weak learning models that are slightly better than random models. It is an ensemble method that generates multiple classifiers from a base learner and ensembles them for building the best classifier. In boosting algorithm, strong classifier is built as an combination of a number of weak classifiers, where each classifier is chosen at every iteration if its accuracy is greater than 50%. At the end of each iteration, the samples are re-weighted in such a way that the misclassified samples get a higher weight so that the next weak classifier shows a better performance on those samples that were misclassified in the previous iteration.

We propose a heterogeneous boosting based integration framework that will exploit all the available (including partial) information from multiple data sources. To achieve this goal, we propose a novel objective criterion which will emphasize the importance of data objects with partial information compared to the common ones. We will also modify the reweighting scheme in the following manner: if a data object is present in only one source out of n sources, the importance of it will be increased by n times while modeling that data object. The increase of weight of the misclassified data object will be inversely proportional to the number of data sources that contain the information about the data object. *The basic intuition here is that the algorithm will give more importance to a data object if it is available only in one source compared to one being available in many sources.* At the end

of each boosting iteration, the instances are re-weighted in such a way that the misclassified objects get a higher weight so that the next weak classifier gives more importance to those objects that were misclassified in the previous iteration. Note that, if it is misclassified in other data sources during an iteration, then its weight is increased in that data source as well. Thus, it is unlikely that a data object is neglected in all the strong models thus making the stronger model more general and diverse.

Let λ denote an indicator matrix such that λ_{ki} is 1 if the data object d_k is in dataset D_i and 0 otherwise. We define $\overline{\lambda_i}$ as the average number of datasets in which the data from D_i is present and is calculated as follows:

$$\overline{\lambda_i} = \frac{\sum_k (\lambda_{ki} * \sum_{i'} \lambda_{ki'})}{\sum_k \lambda_{ki}} \quad (1)$$

We modify the re-weighting scheme in AdaBoost to follow the two above mentioned criteria as follows:

$$\frac{1-\epsilon_j}{\epsilon_j} = 1 + \frac{(1-2*\epsilon_j)}{\epsilon_j} \quad (2)$$

where ϵ_j is the error rate for that iteration. Hence, the increment amount is $\frac{(1-2*\epsilon_j)}{\epsilon_j}$ which is positive since $\epsilon_j < 0.5$. We varied this increment amount based on the number of data sources that the object is present in:

$$\frac{(1-2*\epsilon_j)}{\epsilon_j} * \frac{\overline{\lambda_i}}{\sum_{i'} \lambda_{ki'}} \quad (3)$$

Algorithm 1 HETEROBOOST

1: **Input:** Data sets $D_1 ... D_N$, samples $d_1 ... d_m$ and the indicator matrix λ
2: **Output:** Final stronger classifier M
3: **Procedure:**
4: for $i = 1..N$:
5: Initialize weight $w_{ki} = \frac{1}{m_i}$
6: for $j = 1..T$:
7: (a) $C_{ij} \leftarrow \arg\min_{C_{ij} \in H} \sum_k \lambda_{ki} w_{kj} \, |C_{ij}(d_k) \neq y_k|$
8: (b) for $k = 1..m$:
9: (i) $c_{ij} = log(1 + \frac{(1-2*\epsilon_j)}{\epsilon_j} * \frac{\overline{\lambda_i}}{\sum_{i'} \lambda_{ki'}})$
10: (ii) $w_{ki} = w_{ki} * exp\,[c_{ij} . |C_{ij}(d_k) \neq y_k|]$
11: (c) Normalize weights: $\forall_k \; w_{ki} = \frac{w_{ki}\lambda_{ki}}{\sum_k w_{ki}\lambda_{ki}}$
12: end for
13: strong classifier $M_i(x) = \sum c_{ij} . C_{ij}(x)$
14: end for
15: return classifier $M(d) = \frac{\sum_{i=1}^{n} F_i M_i(d) \lambda_i}{\sum_{i=1}^{n} F_i \lambda_i}$

When the prediction model is used on a particular test case, it is unlikely that the test case will have information for all the sources. Our method will consider only those sources where the information about the test cases is available and then obtain a weighted ensemble model out of those sources. In other words, there will be no imputation performed in the sources that do not have information about that particular gene. The final outcome is calculated as follows: $M(d) = \frac{\sum_{i=1}^{n} F_i M_i(d) \lambda_i}{\sum_{i=1}^{n} F_i \lambda_i}$ where F_i is the evaluation metric (such as accuracy or F-measure) that is being measured for the strong boosted classifier M_i and d is the test case.

4. RESULTS AND DISCUSSION

Since the problem of integrating information from multiple sources naturally occurs in biological domains, we chose different bio-molecular datasets to demonstrate the advantages of the proposed framework. In order to evaluate the effectiveness of the proposed integration framework, we used the genes from S.cerevisiae (yeast) which is the most widely studied model organism for which vast amounts of bio-molecular data are available. The six biological data sources used in our experiments are thoroughly described in [12]. We used functional annotations collected from the Functional Catalogue (FunCat) database [10] to associate each of the genes in the datasets to a functional class. The dataset comprised of 1901 genes that are common across all the data sources and the rest of the 2764 other genes that are present in only fewer data sources. We performed three-fold cross validation for reporting our results on test data. We randomly divided both the common and uncommon genes into three folds. The combination of two folds is used for training and the rest for testing both for common and uncommon genes. To get the combined result, one fold of the common genes is merged with one fold of the uncommon genes to make one fold of the combined genes. For evaluation, we used F-measure metric which is a more appropriate measure for this problem because it is more important to correctly associate the genes with a particular functional class than correctly detecting that the gene is not associated with other function class. Because of the class imbalance problem, the models often produce poor result for F-measure for the target class. To tackle the class imbalance issue, we pre-processed the training data before the training process to under samplify the majority classes and oversamplify the minority class using Synthetic Minority Oversampling TEchnique (SMOTE)[1]. For fairness in comparison, while working with the kernel fusion methods, we used the same fold generated for the heterogeneous boosting. For KF_UMI, we first imputed the missing feature values and then generated complete kernel matrices for different datasets. For the other two methods, we imputed in the kernel matrix to obtain a complete kernel matrix. In either case, we have a set of full kernel matrix weighted summation of those kernel matrices based on the individual accuracy in corresponding datasets.

Table 2 shows the results of different boosting methods on common and uncommon genes. We observe that Ensemble using common genes performs well on the common set of genes. However, for the uncommon genes, the results are not impressive for the ensemble model built on common genes. Using the proposed heterogeneous boosting, we observe the improvement of using all the genes during the training process on the uncommon genes. Finally, *we conclude that using the modified weighting criterion which emphasized the importance for uncommon genes compared to the common genes yields performance improvement for the uncommon genes as well as the overall result.* We also compared our heterogeneous boosting algorithm with kernel fusion methods with different imputation schemes (see Table 3). The result of kernel fusion methods with imputation is inferior to our proposed heterogeneous boosting method. By the use of SMOTE on the training data, which is easily applicable and natural fits the boosting methods, the result of heterogeneous boosting significantly outperformed the kernel fusion method.

Table 2: Comparison of F-measure values in the presence of all the genes using heterogeneous boosting, and boosting with all genes and with only the common genes. E_{CG} - Ensemble with common genes. E_{AG} - Ensemble with all genes. HBOOST - the proposed HeteroBoost method.

Functional class	Common Genes			Uncommon Genes			Overall Result		
	E_{CG}	E_{AG}	HBOOST	E_{CG}	E_{AG}	HBOOST	E_{CG}	E_{AG}	HBOOST
Metabolism	**0.781**	0.779	0.771	0.651	0.735	**0.771**	0.707	0.756	**0.771**
Energy	**0.632**	0.624	0.612	0.478	0.608	**0.631**	0.534	0.613	**0.621**
Transcription	**0.712**	0.690	0.673	0.609	0.683	**0.721**	0.653	0.687	**0.695**
Protein Synthesis	**0.722**	0.692	0.675	0.613	0.685	**0.732**	0.673	0.689	**0.715**
Protein Fate	**0.691**	0.688	0.683	0.591	0.638	**0.706**	0.654	0.664	**0.699**
Protein with Binding Function	0.619	0.622	**0.631**	0.509	0.601	**0.651**	0.559	0.613	**0.645**
Regulation of Metabolism	0.407	**0.445**	0.443	0.391	0.403	**0.441**	0.399	0.425	**0.443**
Cellular Transport	**0.702**	0.690	0.676	0.609	0.653	**0.732**	0.657	0.667	**0.715**
Cellular Communication/Signal	0.515	**0.520**	0.503	0.410	0.453	**0.551**	0.456	0.487	**0.535**
Average	**0.642**	0.639	0.630	0.540	0.607	**0.660**	0.588	0.622	**0.649**

Table 3: Comparison of F-measure values for the proposed heterogeneous boosting method and different kernel fusion approaches

Functional Class	KF_UMI	KF_WSI	KF_NNI	HBOOST
Metabolism	0.531	0.572	0.563	**0.771**
Energy	0.472	0.442	0.515	**0.621**
Transcription	0.482	0.418	0.467	**0.695**
Protein Synthesis	0.509	0.523	0.593	**0.715**
Protein Fate	0.513	0.509	0.567	**0.699**

5. CONCLUSION

The availability of different information about the same data objects created new opportunities as well as challenges for the task of class prediction. The information in heterogeneous sources is often incomplete and most of the recent research on data integration have been primarily focused on the cases where the information is available across all the different sources. In this paper, we developed a new framework that uses all the available information even if some sources do not provide any information about some objects. Our study also shows that boosting the decisions made from individual sources can obtain robust results on predicting gene functions. Furthermore, giving different weights to the uncommon genes helped in improving the predictive ability for the overall classification. We demonstrated the effectiveness of the proposed framework for the problem of gene function prediction and compare to the state-of-the-art ensemble methods using several real-world biological datasets. The proposed heterogeneous boosting method outperformed the standard kernel fusion based approaches for integrating multiple sources in the presence of missing information.

6. REFERENCES

[1] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[2] M. des Jardins, P. Karp, M. Krummenacker, T. Lee, and C. Ouzounis. Prediction of enzyme classification from protein sequence without the use of sequence similarity. *In Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 92–99, 1997.

[3] T. G. Dietterich. Ensemble methods in machine learning. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15, London, UK, 2000. Springer-Verlag.

[4] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[5] U. Karaoz, T. Murali, S. Letovsky, Y. Zheng, C. Ding, C. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl Acad. Sci. USA*, 101:2888–2893, 2004.

[6] L. Kuncheva, J. Bezdek, and R. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[7] G. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, and W. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004.

[8] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[9] F. Roli, G. Giacinto, and V. Gianni. Methods for designing multiple classifier systems. In *Multiple Classifier Systems*, pages 78–87, 2001.

[10] A. Ruepp, D. Zollner, A.and Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H. Mewes. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.

[11] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of International Workshop on Artificial Intelligence and Statistics*, pages 325–332, 2005.

[12] G. Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE ACM Transactions on Computational Biology and Bioinformatics (in press)*, 2010.

[13] X. Zhao, L. Chen, and K. Aihara. Protein function prediction with the shortest path in functional linkage graph and boosting. *International Journalof Bioinformatics Research and Application*, 4(4):375–384, 2008.