Data mining models as services on the internet

Sunita Sarawagi   Sree Hari Nagaralu

Indian Institute of Technology Bombay

sunita@it.iitb.ernet.in

ABSTRACT

The goal of this article is to raise a debate on the usefulness of providing data mining models as services on the internet. These services can be provided by anyone with adequate data and expertise and made available on the internet for anyone to use. For instance, Yahoo or Altavista, given their huge categorized document collection, can train a document classifier and provide the model as a service on the internet. This way data mining can be made accessible to a wider audience instead of being limited to people with the data and the expertise. A host of practical problems need to be solved before this idea can be made to work. We identify them and close with an invitation for further debate and investigation.

1. INTRODUCTION

The Internet has heralded an era of sharing. Vast treasure houses of authoritative information[1] that previously were confined to books with limited accessibility are now available free on the Internet. This proposal is about extending the internet's power of information sharing to knowledge sharing. Anyone with huge amounts of data can host these knowledge servers by building models from their accumulated data on any aspect of decision making. A potential user can consult one or more of these servers and choose from the opinion of these various sites to make their final decision. The mining servers are used in a totally ad hoc, per-user and per-instance basis much like the way documents are accessed on the web. However, while rich document sources have long since found their way to the internet, rich sources of data are still within the confines of disconnected large databases. The few instances of knowledge sharing practised today are all based on model buy-outs. Modulo the problem of communication latency, the service approach is superior to the existing buy-out approach for several reasons. First, an internet service allows greater sharing and accessibility to end users. Second, the user can access the most up to date model. As the server gains new data, its model can be refined and the latest results available to the user right away. Other advantages are reduced overheads of software installations, increased user mobility and reduced cost to the occasional user of the software. The downside is lack of sufficient bandwidth to all locations and confidentiality of data or model.

The current wave of Application Service Providers (ASPs)[2] that offer business solutions and applications on the internet, make the idea of a mining service provider all the more relevant. Existing ASPs already cover a broad range of rentable functions, including desktop management, storage management and ERP systems. In addition, a growing number of mining companies are starting to provide business intelligence solutions as internet application services[3]. The basic approach followed by such companies is to retrieve the customer data from their operational system, in some cases augment it with relevant demographic data available from the company and provide the mined knowledge as a web interface for the client. The mining service model proposed here is different in that it involves sharing not simply of software and skills but also data and model. Also, the usage model is more ad hoc and does not require a priori agreements between the service provider and the user. A closely related project is the MMM project [5] where the goal is to share economic models on the internet much along the lines that we are proposing here. Another related concept is exemplified by the several "Ask an expert"[4] sites prevelant on the net. These are however backed by human experts who answer the questions posed by users either free or at a fee. Our goal is to automate these through mining models.

The goals of this project overlap with the goals of recent work in the area of distributed data mining [6] and the closely related area of multi-agent learning [14; 15]. There are several projects underway on large scale distributed data mining: some examples are the Kensington project [2] for mining enterprise data distributed across the internet, the Papyrus project [4] for providing a high performance networking and computing testbed for mining on the internet, the JAM project [12] for developing a java agent based meta learning framework for distributed mining and the BODHI project [7] for doing collective data mining with stress on learning from vertically partitioned data.

The field of distributed data mining shares some of the same concerns as web mining services especially in the area of standardization and model integration. However, there are significant differences. First, almost all the above distributed mining projects focus on model construction in a distributed fashion followed by integration. Consequently, the stress is on protocols and software architectures for exchanging data and partial models. We are agnostic about

[1] `http://dir.yahoo.com/Reference/`

[2] check out `http://www.aspstreet.com/`, for instance

[3] `http://www.kdnuggets.com/solutions/asp.html`

[4] `http://dir.yahoo.com/Reference/Ask_an_Expert/`

how model construction happens — that is something that individual knowledge servers have to worry about before putting their models as services on the web. For model integration they all propose some variation or enhancement of the basic meta learning algorithms that require a single coordination phase and a separate validation set. We elaborate on this issue in Section 2.3. Finally, these are not meant for use by an ad hoc internet user but rather assume a closed pre-defined set of participants.

1.1 Example scenarios

We present some example scenarios of using internet knowledge servers.

1.1.1 Document classification services

An imminently useful mining service is a document classification service that accepts documents and predicts its class from a predefined category tree. The internet has several large categorized document sources — some examples are Yahoo[5] and Altavista[6] for general web documents, CoRR[7] and NCSTRL[8] for computer science publications; PubMed[9] for medical pulications and so on — the list is endless. These data rich sites can easily use their stored, categorized data sets to build an automatic text classifier. The model can then be made available as a service where users can submit their documents and get back its position in the document taxonomy used at the site. Consider a new digital library or newspaper agency that wants to automatically categorize its submissions on a standard taxonomy. Instead of downloading the huge amounts of data on its site and spending money and effort in building a good automatic classifier, the agency might be willing to use the categorization service even if it involves a fee.

The problem that the user will need to contend with is choosing from the predictions of different portals if they assign different categories to the same document. Which prediction should be trusted more? Presumably different sites have their areas of strength and weakness and no one single site will be constantly better than the other. Can he cascade different classifiers? For instance, use Yahoo for a coarse grained prediction of whether a document is about computer science or medicine and then use either CoRR or PubMed to do a more detailed classification. We discuss these and several other interesting research issues in Section 2.

1.1.2 Collaborative filtering service

Another compelling application scenario arises from electronic commerce. Consider a new store that wants to sell streaming movies on the internet. The store would like to provide meaningful, personalized recommendations to its customers based on their demographic profile or preference patterns of other similar users. A new store like this one, however, does not have enough data to learn these patterns to start with. It might want to use the services of other mining models. For example, IMDB[10], an internet movie database offers recommendation on similar movies and so do a few other sites like Movie Critic[11] and Each Movie[12]. These sites keep getting updated and instead of negotiating for a copy of their preference database, the new store could simply use the services of the recommendation model online. Similarly, Amazon's book database can provide the relationship between a person's profile and the preferred book genre ("science fiction", "romantic", "philosophical" etc). A similar classification applies to movies and therefore Amazon's collaborative filtering system on books can be used to make coarse grained predictions on the type of movie. This can be followed by accessing a collaborative filtering engine on movies and actors themselves for further refinement of the recommendation. Amazon's collaborative engine perhaps was trained on larger amounts of data and therefore even though the domain is not the same, the coarse-grained prediction on the genre of the movie is likely to be better. Also, different sites might be good at predicting for different subsets of population. Filtering on Indian movies is perhaps best done through an Indian site and filtering on American movies is best handled by an American site. Here again the problem is how to combine predictions across similar models, across models at different levels of granularity and partially developed models.

Eventually, as the store collects more and more of its own sales data, it might want to incorporate that experience into the recommendations. The store could now build its own recommendation engine. But, it may not have the expertise or the infrastructure. Also, doing so will deprive it of the experiences of other stores. A better option is such cases, could be to ship its data to the service provider which then figures out how to best combine the predictions of the store's data with the previous models.

1.1.3 Risk prediction services

Consider another scenario, this time involving data from traditional businesses. A new insurance company wants to draw upon the experiences of existing businesses in determining rates and charges for new applicants. A key factor in determining premium rates is the the risk level of the applicant — Is he likely to engage in insurance frauds? Will he pay his premiums regularly? The new insurance company has no historical data for building a model for answering these questions. Another insurance company in the same business may not want to share its model. However, a credit card company not in the insurance business might want to sell the data to the new company. Although, they are in different businesses, basic data about the credit worthiness of an applicant is the same in both cases and can be easily shared[13]

This scenario is different from the previous two in that there are stronger confidentiality concerns here. In this case, perhaps the two parties would go into a separate agreement before sharing the models and that too on secure private networks. However, some of the same concerns arise. The insurance company might want to use models of multiple credit card companies covering different geographical regions. With every new applicant the company needs to decide

[5] http://www.yahoo.com

[6] http://www.altavista.com

[7] http://xxx.lanl.gov/new/cs.html

[8] http://www.ncstrl.org/

[9] http://www.ncbi.nlm.nih.gov/PubMed/

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

[11] http://www.moviecritic.com

[12] http://www.eachmovie.com

[13] This notion of risk assessment is very different from existing ratings assigned by credit bureau agencies on specific individuals.

which prediction to choose. The approach might differ based on whether geography influences the risk level or not. But, how can a new company figure that out? Again, as the company collects more and more of its own data it might want to somehow integrate that data with the external data of the provider.

2. RESEARCH CHALLENGES

There are several challenges in the way of making ad hoc mining models on the web a reality. We list some interesting ones here.

2.1 Standardization

Standardization of data and model is fundamental to the effective deployment of combined collaborative models across distributed internet applications. Document sharing is easier because text documents are mostly self-describing and human interpretable. In contrast, data even with schema definition is too hard to interpret outside an established user community. This is slowly but gradually changing thanks to the business-to-business E-commerce industry. There is a growing number of consortiums on standardizing various aspects of day-to-day business information (See ebXML[14] for example). Even in mining there is increasing move towards standardization — at least for some well understood domains like model prediction. Some prominent examples are OLE DB for Data Mining [15] proposed at Microsoft, CRoss-Industry Standard Process Model for Data Mining (CRISP-DM)[16] and the Data Mining Group[17]'s Predictive Model Markup Language (PMML) based on XML.
There are several aspects of standardizations: the first step is standardization at the syntactic level of the input/output formats. This aspect is addressed at various levels by the low-level RPC protocols like SOAP[18] and by the various mining standards mentioned above. The second aspect is standardizing semantics of data i.e., identifying attribute names and their meaning within a particular industry group. Several vertical industry groups are already attempting this in the XML context. A third issue is standardizing the model structure. For instance, in the case of hierarchical classifiers, this involves agreeing on the structure of the tree. Such an agreement is harder. Yahoo and Altavista, even though based on the same data source have different taxonomies of their web directories. Finally, providing a method for expressing the capability or scope of each model whereby it can specify what part of the data it was trained on, i.e., specify the coverage of the training data.

2.2 Confidentiality of data and model

Confidentiality is another major concern in sharing of data or models on the internet. Most large data sources are behind the zealous guards of company firewalls. Two phenomenon are promising to remove this limitation. First, new data sources of popular appeal are increasingly made freely available on the web. This holds strongly for the document classification scenario presented earlier and partially for the collaborative filtering scenario. Second, better business to business security infrastructure provides greater contol to the provider on who can use their model or data. Also, sharing of summarized models raises fewer confidentiality concerns than sharing of raw data. For instance, in the insurance scenario presented earlier, a credit company would perhaps be willing to share its risk prediction model with the insurance company but not the raw data. Thus, even the previously firewalled datasets could be made available for sharing albeit under strict access control. Finally, there are particular applications like fraud detection where even competitive banks are willing to share their data [12]. Another concern is confidentiality of the user's data. For example, in the document classification scenario although the model is freely sharable, the user deploying the model might have security concerns about shipping their data to the service provider. Is there any way of convincing a user that the service provider would not log his data on the side?

2.3 Integrating distributed models

Eventually multiple sites will start offering prediction services on the same domain. For instance, for document classification a user could go to Yahoo or Altavista or any of the other sites that serve categorized web documents. In the Movie recommendation example, the user could choose from IMDB or Movie Critic. Each of these sites cover partially overlapping data sets and the user is left with a decision to make on which one. We briefly review the classical work on meta-learning [1] first and later discuss why these do not apply in our context.

2.3.1 Meta-Learning:

Meta learning refers to the method of integrating the results of the classification of several component models that are trained independently. A variety of different methods for constructing meta models from individual models learnt in parallel have been proposed ([10] presents a survey). Based on the method used for classification these can be broadly classified as follows:

- **Voting:** that adds the votes of different classifiers on a class and chooses the one with the highest vote.
- **Arbiter:** that uses an arbitration rule for choosing between classifiers when they cannot reach a consensus
- **Combiner:** that explicitly train a new meta classifier on the predictions of the component classifiers using a validation data set. The meta learner can be of various types depending on the set of attributes used for meta learning. On the one extreme are meta-learners that use only the class predictions of the component models for training and on the other extreme are those that use both the class predictions and all the original input attributes — these are also called augmenters.

The first two methods Voting and Arbiter are too simplistic and have been shown to be inferior to the Combiner methods. These methods are not suitable for our purpose for several reasons. First, the participating sites are autonomous which means the input data, method and time of model construction proceeds autonomously without any coordination with other sites. Second, sites might be constantly evolving and changing their model at will. A knowledge server may not even know about the existence of some other knowledge

[14] `http://www.ebxml.com/`
[15] `http://www.microsoft.com/data/oledb/dm.htm`
[16] `http://www.crisp-dm.org`
[17] `http://www.dmg.org/`
[18] `http://www.w3.org/TR/SOAP/`

server. Therefore, traditional model selection approaches that use a single off-line training phase for combining the different models are not applicable. Finally, even the type of users of the model are different. Unlike the trained mining specialists, we will now have more one-time occasional users. Such users are not likely to have historical data to benchmark the models and choose amongst them. Therefore, existing methods of meta learning [9; 10; 3] and knowledge probing [2] that all require a separate validation set are ruled out. We now list some research areas that arise when trying to integrate the output of several mining models.

2.3.2 Dynamic model selection

The first problem is designing a dynamic model selection algorithm that does not require any global synchronization between sites, does not rely on the availability of a validation set from the user and maintains full autonomy of the participating sites in terms of what prediction model they use and how and when they change their model. An obvious algorithm that meets all of these criteria is the majority voting algorithm. Voting works well sometimes in a homogeneous setting but will fail when sites are specialized to be particularly strong in one topic compared to all other sites. For instance, although there are several US-based movie recommendation sites, for a Chinese movie, the prediction of a site specializing in that topic should be preferred. Another strategy is to let each classifier output additional information about its confidence in making the prediction. There has been some research work [11; 13] in this direction but none of these take into account the difference in the coverage of the different models.

2.3.3 Using models at various levels of granularity

How can we adapt the above dynamic model selection problem when class labels are arranged in a hierarchy like in the case of Yahoo? In such cases, more interesting integration of models can be done by using different models for different levels of the taxonomy. In our text categorization scenario we could use sites like Yahoo to do the first coarse level categorization of whether a document belongs to "Computer science" or not and then use a special CS repository to provide further subject level classification. A research issue here is how to decide the level up to which we should use Yahoo's classification.

2.3.4 Composing from partial models

In the previous two examples, we assumed that there is only one correct class label for an instance. What if an instance, could have more than one class label. For instance, a paper presenting an application of data mining in medicine is both about data mining and medicine. A site like CoRR that can only classify papers based on a taxonomy of computer science subjects would assign a class label of "Data mining" and maybe Yahoo would assign it two class labels: "Medicine" and "Mineral mining". While the first class is correct, the second class is a mistake. How can a model selection method handle such cases?

2.3.5 Vertically partitioning of data

Another situation is where the input attributes are vertically partitioned across different sites. For instance, data about a student could be vertically partitioned across two databases. The institution might have stored academic records of the student and the alumni database could have kept the track record of the student since graduation. If we wish to study a model that can predict a student's performance in the real world based on their academic records and initial placements, we would need to combine models from these two vertically partitioned data sources. Some initial attempts have been made to learn from vertically partitioned data [7] but these have the same problem of requiring centralized coordination and synchronization as the meta learning methods. The problem is still open for further research and investigation.

2.4 Personalizing a mining model

After using a mining service for a sufficient amount of time, a user along the way would have collected his own data about the performance of the model. At this point, he could either train his own model that works best for his data or provide his data along with their true labels as feedback to the mining model. The second approach could be preferred for two reasons: first, the user could continue to benefit from the data of the provider and second, he may not have the expertise and infrastructure to build his own model. The user might want to enter into a special contract with the service provider to build a special model personalized for his data and also including the provider's data whenever the user data is insufficient. How does the provider build such a model? How much of the provider data to include so as not to overshadow the user's data but at the same time help him build a more robust and complete model whenever possible? When calibrating the accuracy of such a model should only the user's data be taken into account? That data may not be big enough and may not cover all interesting cases as seen in the entire data. A similar problem arises when capturing drift of a model as time progresses [8; 16]. What proportion of the data should be recent versus old? A standard approach is to assign an aging factor and weight each instance by that factor. It is not clear how to choose this aging factor and if all instances should age at the same rate.

3. REFERENCES

[1] P. Chan, S. S., and D. Wolpert, editors. *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models*, Portland OR, 1996. AAAI.

[2] J. Chattratichat, J. Darlington, Y. Guo, S. Hedvall, M. Khler, J. S. A. Saleem, and D. Yang. Deploying enterprise data mining on the internet. In *PAKDD*, 1998. `http://ruby.doc.ic.ac.uk/`.

[3] P. Domingos. Knowledge discovery via multiple models. *International Data Analysis*, 1998.

[4] R. L. Grossman, S. Kasif, D. Mon, A. Ramu, and B. Malhi. The preliminary design of papyrus: A system for high performance, distributed data mining over clusters, meta-clusters and super-clusters. In Kargupta et al. [6]. `http://www.lac.uic.edu/~grossman/cv/dataspace-background.htm`.

[5] O. Günther, R. Koerstein, R. Krishnan, R. Müller, and P. Schmidt. The mmm project: Access to algorithms via www. In *Poster presented at the Third International*

World-Wide Web Conference, Germany, 1995. `http://macke.wiwi.hu-berlin.de/mmm/`.

[6] H. Kargupta et al., editors. *Workshop on Distributed Data Mining, The Fourth International Conference on Knowledge Discovery and Data Mining*, 1998. `http://www.eecs.wsu.edu/~hillol/kdd98ws.html`.

[7] H. Kargupta and B. Park. The collective data mining: A technology for ubiquitous data analysis from distributed heterogeneous sites. *Submitted to IEEE Computer Special Issue on Data Mining*, 1998. `http://www.eecs.wsu.edu/~hillol/ddm.html`.

[8] M. G. Kelly, D. J. Hand, and N. M. Adams. The impact of changing populations on classifier performance. In *Proceedings of Fifth International Conference on SIG Knowledge Discovery and Data Mining (SIGKDD)*, 1999.

[9] P.Chan and S.Stolfo. Toward parallel and distributed learning by meta-learning. *In Proceedings of the Second International Workshop on Multistrategy Learning*, pages 15–165, 1993.

[10] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. *In Proc. of the 3rd Int'l Conf. on Knowledge Discovery and Data Mining*, 1997.

[11] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[12] S. Stolfo, A. Prodromidis, S. Tselepis, W. Lee, D. Fan, and P. Chan. JAM: Java agents for meta-learning over distributed databases. *In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997. `http://www.cs.columbia.edu/~sal/JAM/PROJECT/`.

[13] A. S. Weigend and D. A. Nix. Predictions with confidence intervals (local error bars). *ICONIP, Seoul*, 1994.

[14] G. Weis, editor. *Distributed artificial intelligence meets machine learning: Learning in Multi-Agent Environments.* Springer Verlag, 1996.

[15] G. Weiss, editor. *MULTIAGENT SYSTEMS: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.

[16] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23, 1996.