An Ensemble-based Recommendation Engine for Scientific Data Transfers

William Agnew
Georgia Inst. of Tech.
30332 North Ave NW
Atlanta, GA
wagnew3@gatech.edu

Michael Fischer
University of
Wisconsin-Milwaukee
Milwaukee, WI
fisch355@uwm.edu

Ian Foster
University of Chicago
5801 S Ellis Ave
Chicago, IL
foster@anl.gov

Kyle Chard
University of Chicago
5801 S Ellis Ave
Chicago, IL
chard@uchicago.edu

ABSTRACT

Big data scientists face the challenge of locating valuable datasets across a network of distributed storage locations. We explore methods for recommending storage locations ("endpoints") for users based on a range of prediction models including collaborative filtering and heuristics that consider available information such as user, institution, access history, endpoint ownership, and endpoint usage. We combine the strengths of these models by training a deep recurrent neural network on their predictions. Collectively we show, via analysis of historical usage from the Globus research data management service, that our approach can predict the next storage location accessed by users with 80.3% and 95.3% accuracy for top-1 and top-3 recommendations, respectively. Additionally, our heuristics can predict the endpoints that users will use in the future with over 75% precision and recall.

1. INTRODUCTION

As data volumes and network speeds increase, the task of determining where useful data are to be found becomes more complex. Services such as Globus [12] simplify the management of scientific data (for example, by streamlining sharing [8] and publication [7]). Still, an individual scientist may have access to hundreds or even thousands of storage systems. Which should they visit next?

Commercial web services, such as travel, e-commerce, and television streaming services rely on user-specific recommendations to both enhance user experience and drive revenue streams [4]. These recommendations are possible because of the large amount of usage information captured by these services. Here we investigate the feasibility of providing similar, targeted data location recommendations to scientists, with the goals of improving both 1) user experience and 2) understanding of how large scientific data are used. We use the approximately 3.5 million transfer operations conducted via Globus between research storage systems over the past five years as a basis for this study. We envisage such capabilities could be offered as an online recommendation engine that would allow users to quickly find the data they are looking for while also enabling them to explore other data of relevance (Figure 1).

We define and evaluate a collection of specialized endpoint prediction heuristics that consider unique features of large scientific data, Globus users, and storage endpoint information (e.g., institution, transfer frequency, and endpoint location) derived from historical Globus usage data. We measure the performance of these heuristics in terms of how well they predict 1) the two specific endpoints used in a user's *next* transaction and 2) the set of endpoints used by a a user in the *future*. We show that we can predict the *next* endpoint correctly over 95% of the time and *future* endpoints with over 75% precision and recall. In addition, by analyzing the relative contributions of the different features used by the heuristics, we explore the contribution of each feature to our recommendations. We find a large and surprising difference between good recommendation strategies for source and destination endpoints.

The rest of this paper is as follows. In §2 we investigate historical Globus usage as the basis for developing endpoint recommendation heuristics. We then describe in §3 a baseline recommendation algorithm, using industry standard collaborative filtering. In §4 we describe a series of endpoint recommendation heuristics developed using our observations of historical usage. In §5, we present the neural network used to combine our heuristics' recommendations. Next, in §6, we evaluate the performance of our approaches. Finally, we compare with related work in §7 and conclude in §8.

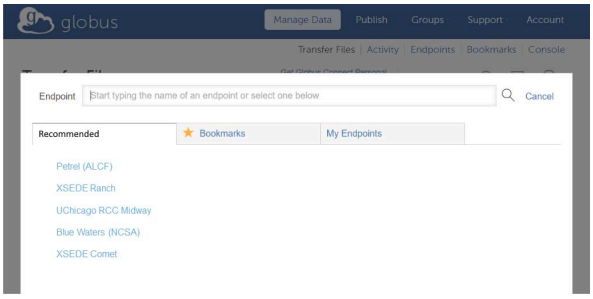


Figure 1: Mockup of recommendation interface.

978-1-5090-6158-7/16 $31.00 © 2016 IEEE
DOI 10.1109/DataCloud.2016.5

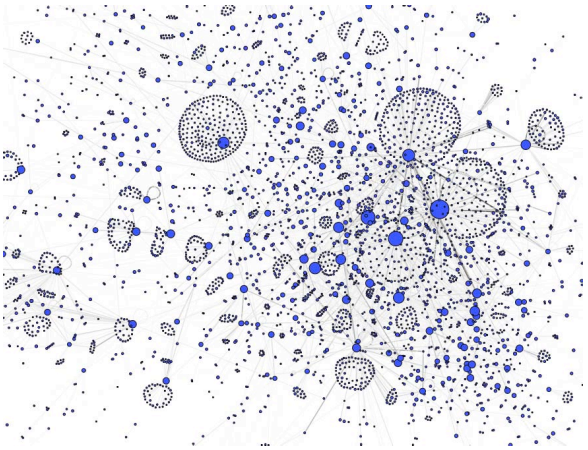


Figure 2: Globus Network. Endpoints are represented as vertices. Edges represent transfers between endpoints. Larger endpoints have transferred more frequently with more distinct endpoints. Endpoints that transfer more frequently with each other are closer.

2. GLOBUS

Globus, a software as a service provider of research management capabilities, supports data transfer, synchronization, and sharing [2]; data publication [7]; identity management, authentication, and authorization [21]; and profile and groups management [6]. It provides a rich source of information from which we can understand scientific data access patterns. Thus, as the basis for developing recommendation heuristics we first explore historical Globus usage to derive features that may be indicative of usage.

Over its six years of existence, Globus has been used to conduct almost 3.5 million transfers, totaling more than 180PB and 2.5 billion files, among 23,000 unique endpoints. Thus, Globus usage can be represented as a network with 23,000 vertices and 3.5 million edges. Part of this network is illustrated in Figure 2.

The graph highlights the different usage patterns of Globus users and endpoints. There are distinct endpoint clusters, typically centered around a single large (more frequently used) endpoint. These clusters are clearly related in some way, perhaps, for instance associated with a data source (e.g., the National Center for Atmospheric Research's Research Data Archive), a particular scientific group, or a particular instrument or resource (e.g., the BlueWaters Supercomputer). Some clusters are completely independent of the rest of the network, while others are more tightly connected. We also see that a small number of endpoints have been involved in many transfers with many endpoints, while many endpoints have participated in few transfers with few endpoints.

Globus is used in a broad range of scenarios, including automated usage by scripts and third-party applications. Due to our primary interest in providing recommendations to users, rather than programs (e.g., those that backup supercomputers), that use Globus, we focus our prediction efforts on the roughly 800,000 operations initiated using the web interface. Figure 3 further illustrates usage patterns showing the data volumes, number of transfers, and number of unique endpoints per user for all Globus usage submitted via the Web interface. These long-tailed distributions are a defining feature of Globus (and perhaps scientific data usage in general), and present a challenge for endpoint prediction. The endpoints with low usage provide little historical information on which to base predictions, and the endpoints with high activity are often used by many different scientists each with different usage patterns.

Although Globus has been used to transfer billions of files, the mappings between users and endpoints, and endpoints and endpoints, are sparse. Only ∼0.01% of potential user/endpoint pairs and 0.006% of potential endpoint/endpoint pairs are present. Figure 4 illustrates this sparsity with a heatmap showing the transactions between the 200 most active users and endpoints.

In addition to highlighting sparsity the heatmap shows two interesting patterns. First, an approximately diagonal line that shows correlation between the most active users and the most active endpoints: that is, active users strongly favor a single endpoint so much so that this favored endpoint has a usage ranking similar to the user's. This striking pattern leads us to suspect that recommending the endpoint most frequently used by a user would provide good results. The second usage characteristic is the presence of vertical lines. While most users use one endpoint much frequently than others, these lines indicate that some endpoints (primarily the most active endpoints) are used by many users. This knowledge can be leveraged by identifying and recommending these broadly used endpoints.

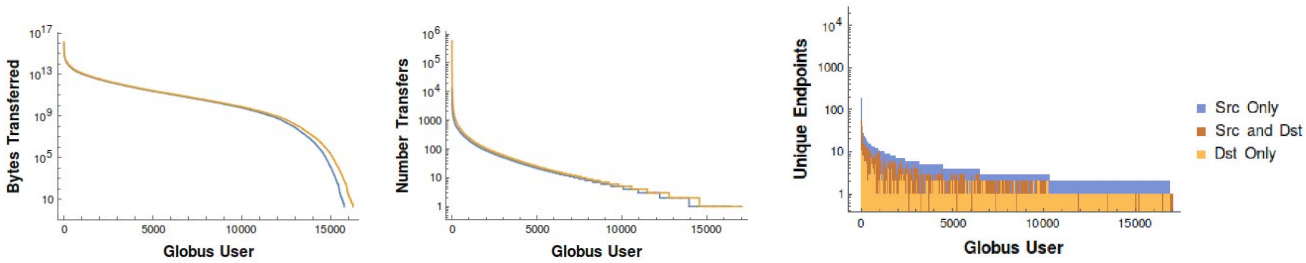
3. COLLABORATIVE FILTERING

Collaborative filtering (CF) is a technique commonly used by recommendation systems to determine rankings based on other users' rankings. In simple terms, the model assumes that if user A has a preference for the same item as user B, then A is more likely to choose another item preferred by B than a user chosen at random. CF techniques are commonly applied to product and movies recommendations: CF was used to win the Netflix Challenge [28], for example. CF has also been used with success to recommend web services to users [27].

While CF is typically used to recommend new products to users based on explicit rankings, thus it is more suitable for our second recommendation task of predicting a set of endpoints to be used in the future. We use the popular GraphLab [19] toolkit to implement a CF model. We set a user's rating of an endpoint to 1 if that user has used that endpoint, and 0 otherwise. We also give the model the parts of users' email suffixes (for example, wagnew3@gatech.edu belongs to the categories "gatech" and "edu") and the owners of each endpoint, the idea being that the CF model could find groups of collaborating users. We then applied the ranking factorization recommender [1], which writes all user-item recommendation weights as an equation of many unknown variables and then uses stochastic gradient descent to find the values for those variables that minimize some cost function. These variable values are then used to predict future endpoint ratings for users.

4. ENDPOINT PREDICTION HEURISTICS

Globus stores detailed records regarding users, endpoints,



(a) Transfer volume per user (b) Transfers per user (c) Distinct endpoints per user

Figure 3: Long-tailed distributions of Globus usage. In (a) and (b), all transfers are in orange, and web (human) initiated transfers are in blue. (c) shows endpoint usage for all transfers; web transfers are similar.

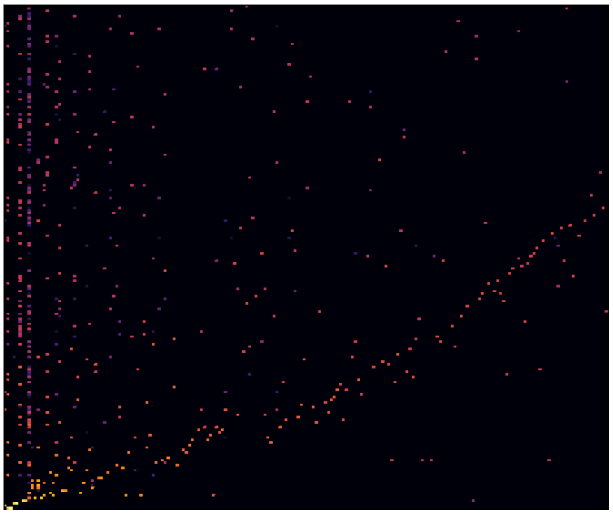


Figure 4: Heatmap of user-endpoint transfer pairs for the 200 most active users and endpoints. Users on y axis. Endpoints on x axis. Colors are based on the log of the number of transactions for each user-endpoint pair.

and transfer history: user names, email addresses, and institutions; endpoint descriptions, locations, and settings; and transfer settings, performance, and errors. We use this information to develop a collection of specialized endpoint recommendation heuristics. When queried with user ID, date, and a positive integer n, each heuristic returns what it believes are the top-n best endpoint recommendations for that user ID on that date. Now we describe each heuristic.

History: The history heuristic does exactly what one would expect: it predicts that the top-n best source (S) / destination (D) endpoints are the n most recently used S/D endpoints.

Markov Chain: The Markov Chain heuristic correlates previously used endpoints with potential future endpoints. To do so, it maintains a transition matrix of the probabilities of using each endpoint as a S/D conditioned on a particular endpoint being previously used as a S/D. These probabilities are estimated online by the Markov chain heuristic from the observed transitions. According to this heuristic, the top-n most likely S/D endpoints for a user are the top-n most likely endpoint transitions given that user's previous S/D endpoint choice.

Most unique users: The most unique users heuristic takes advantage of the long-tailed usage distribution: a small number of endpoints are used by many users and most endpoints are used by few users. The top-n best S/D endpoints are the endpoints with the nth most unique users who used that endpoint as a S/D.

Institution: The institution heuristic maps users to their institution based on that user's associated email suffix. For example, "wagnew3@gatech.edu" would be mapped to the institution "gatech.edu." The top-n best S/D endpoints for a user are the n endpoints owned by a user belonging to the same institution with the most unique users that have used them as a S/D.

Endpoint ownership: The endpoint ownership heuristic recommends the top-n most recently created endpoints owned by a user, based on the idea that if a user creates a new endpoint, that user is likely to use it soon.

5. COMBINING HEURISTICS

Our heuristics model different aspects of usage and therefore perform well for different classes of users. To provide the best possible recommendations we combine these heuristics into a single, superior heuristic. To do so, we trained a deep recurrent neural network [14] on historical Globus data to select the best endpoint recommendations from each heuristic. We choose to use a recurrent neural network over more traditional, simpler ensemble methods because of the great success recurrent neural networks have achieved in learning series [16].

The model for this neural network is shown in Figure 5. The basic model is composed of two LSTM blocks stacked on top of two fully connected layers. The first LSTM block has an output size of 15 and the next LSTM block and the two fully connected layers have input and output sizes of 15. The first fully connected block uses a ReLU activation function which has proven effective in deep neural networks [13]. The last layer of the network uses a softmax activation function, which creates a probability distribution of outputs. We do not claim that this network architecture is optimal, but we justify its complexity by comparing it to a simpler architecture of a single dense layer of size 15 and a softmax output layer (§6).

As we see in Figure 3(c), there is a a significant difference between endpoints used as sources and endpoints used as

destinations. To address this difference we have trained separate neural networks for source and destination endpoint recommendation. We explore this choice in §6.

5.1 Neural network input

The input to the neural network consists of two parts: recommendation weights and additional problem information. To obtain recommendation weights, each heuristic recommends its top-n endpoints, along with their weights. If an endpoint has already been recommended by another heuristic, then the heuristic continues recommending top endpoints until it has recommended n endpoints distinct from the other recommendations or until it can no longer make recommendations. Disallowing duplicate recommendations allows the network to consider more endpoints for its size, but it may reduce accuracy in some cases: if a particular endpoint is weighted highly by many heuristics, then that endpoint is likely a good recommendation. To alleviate this problem, we include each heuristic's weight for each recommended endpoint.

To give a brief example of how we incorporate the heuristics into our neural network, consider an endpoint X. Each heuristic weights endpoint X, that is, outputs how confident it is that endpoint X is the correct endpoint choice. If the user owns only Y, then the Endpoint Ownership heuristic will assign Y a weight of one. If endpoint Z is the second most recent endpoint used, then the history heuristic will assign Z weight of $\frac{1}{2}$. Each heuristic's weighting is fairly arbitrary, having only the property that more highly ranked endpoints have larger weights; in most cases, we use a weighting function of $\frac{1}{rank}$. The neural network is then presented with the set of endpoints X, Y, Z along with their weights. use such arbitrary weighting functions in the absence of more natural weighting functions and leave it to the neural network to learn how to interpret each heuristic's weightings.

The performance of different heuristics could be affected by external information, for example, whether the user works in academia or industry, if the endpoint is located in the same country, or if the data accessed is large. In short, any number of factors could affect endpoint selection. To further improve recommendation accuracy we provide the neural network with every piece of information available and let the neural network learn which information is important. The complete list of additional input information is as follows: transaction date, user institution type (academic, commercial, government, other), number of Globus users affiliated with user institution, total number and volume of user transactions, data size, transaction frequency and volume for each recommended endpoint, current accuracy for each heuristic for the user, and the correctness of each of the previous recommendations made for the user. Finally, each heuristic also provides its overall confidence for its recommendations for that user, again a fairly arbitrary weighting that is interpreted by the neural network. For example, the history heuristic uses the size of the user's transaction history to compute its confidence.

5.2 Neural network output

Recall that we focus on two recommendation use cases: recommending the *next* endpoint used and recommending endpoints that might be used in the *future*. When recommending the next endpoint, we want the neural network to output a weight of 1 if the endpoint will be used as the next transaction and a weight of 0 for all other endpoints. When recommending the set of future endpoints, the desired output is 1 if the corresponding endpoint will be used within the specified time interval after the transaction, and 0 otherwise.

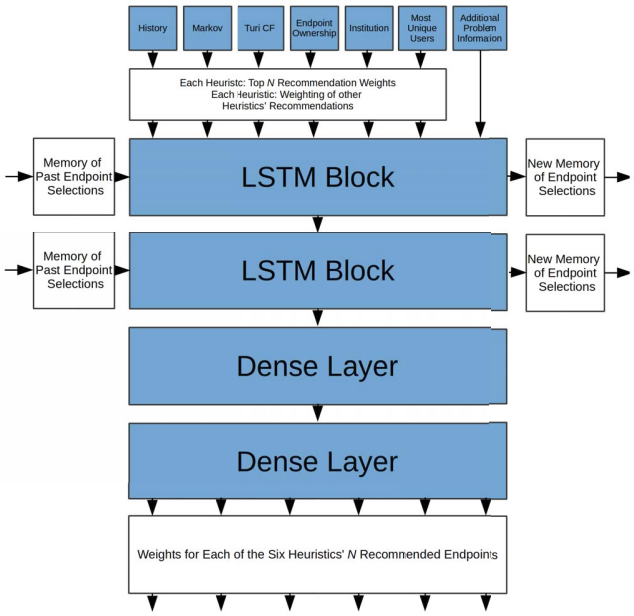


Figure 5: Neural Network Block. Takes as input heuristic recommendation weights and memory from past recommendations to the user, and outputs reweighed endpoint recommendations and updated recommendation memory.

6. EVALUATION

We study our ability to predict 1) the next endpoint used; and 2) the endpoints to be used in the future. In the first case we compare the performance of collaborative filtering, our heuristics, and the ensemble neural network. In the second, we compare the performance of collaborative filtering and the ensemble neural network.

We compare performance with respect to accuracy as well as precision and recall. Accuracy measures our ability to correctly predict endpoints amongst a group of recommendations. We define two accuracy measures. The first, total accuracy, is the fraction of all endpoints correctly predicted where t_p is the number of correct predictions and n_t is the number of total transfers:

$$Total\ Accuracy = \frac{t_p}{n_t} \tag{1}$$

As shown in §2, most users perform few transfers. While predicting endpoints for the highly active users is important, we cannot ignore users with few transfers, as the latter are likely would most benefit from good endpoint prediction. Therefore, to gauge how well our recommendations perform for each user, we also report user accuracy as an average accuracy across all users. Let t_u be the number of transfers

performed by a user and n_u be the total number of users. That is:

$$User\ Accuracy = \frac{\sum_{i=1}^{n} \frac{t_p}{t_u}}{n_u} \quad (2)$$

Precision and recall, two commonly used metrics in recommendation and information retrieval systems, measure how good recommended items are and how likely good items are to be recommended, respectively. They are defined as follows, where t_p and t_n are true positives and negatives, respectively, and f_n is false negatives:

$$Precision = \frac{t_p}{t_p + f_p} \quad (3)$$

$$Recall = \frac{t_p}{t_p + f_n} \quad (4)$$

All training and recommendation algorithms took only a few hours to run on hundreds of thousands of transactions, making them very feasible for real world use.

6.1 Training

We split historical Globus data into training and validation sets. We trained two groups of four (one for each combination of source/destination and top-1/top-3) networks. The first group is used to predict the endpoints to be used in the next transaction, the second group is used To predict the set of endpoints to be used in the future. The networks for predicting endpoints to be used in the next transaction were trained on the first 500,000 transactions for 1000 epochs with batch sizes of 5,000 using the categorical logarithmic objective function, and validated using the most recent approximately 450,000 transactions. For the networks to predict the endpoints used in the future, we first defined "future" for three different time spans: one week, one month, and one year. That is, when we predict endpoints to be used in in the future, we evaluate by predicting a set of endpoints that a user will use in the next week, month, or year. The future endpoint networks were trained for 1000 epochs with a batch size of 5,000 on all but the transactions in the past 1.5 years (about 380,000) with the categorical logarithmic objective function. For validation data, we used every transaction that occurred within the past 1.5 years up until the specific time period evaluated. That is, given the need to evaluate within a time period (e.g., 1 year) we must have at least that period's data available for validated (e.g., for 1 year experiments we validated historical data from 1.5 years ago to 1 year ago). All experiments contained at least at least 150,000 transactions.

6.2 Next endpoint recommendation

We next examine our heuristics' performance at predicting the endpoints to be used in the next transaction. Figure 6 shows each heuristic's total accuracy and Figure 7 shows user accuracy. Each figure shows accuracy when recommending the top-1 and top-3 endpoints. In addition to each heuristic and the neural network, we also report on recommendations produced by a simple neural network implementation (*Shallow Neural Network*) and the optimal algorithm, *Combined*, that selects the correct endpoint if it is predicted by any single heuristic. The shallow neural network contains a single dense layer of size 15 and a softmax output layer.

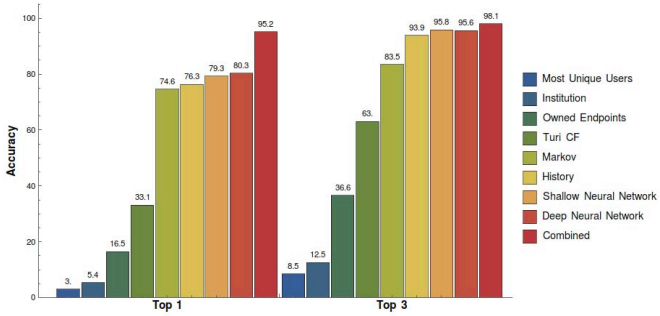


Figure 6: Transfer recommendation accuracy

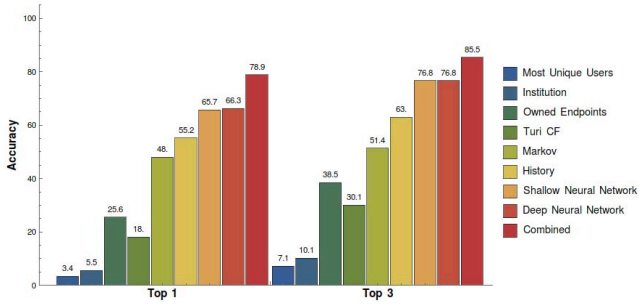


Figure 7: User recommendation accuracy

Figures 6 and 7 show that the *deep neural network* outperforms any single heuristic and indeed performs close to optimally, with a difference of 14.9%/0.8% for transfer recommendation accuracy and 13.7%/5.1% for user recommendation accuracy, relative to the optimal *combined* strategy. The history *heuristic* outperforms the other heuristics in all cases, this is due to the frequency with which users use the same endpoints. The *most unique users*, *institution*, and *owned endpoints* heuristics perform significantly worse than the others. Surprisingly, the *Turi CF* heuristic does much worse than far simpler heuristics, like *history*, providing further evidence of the need for domain-specific knowledge. (In the case of the *history* heuristic, this knowledge may be that users often choose the same endpoint multiple times; in contrast, customers rarely buy the same book multiple times.) It should also be noted, that predicting the same item and using implicit ratings is not a typical use-case for CF. For all heuristics, user accuracy is lower than transaction accuracy, but this is not unexpected: there are many users with little transaction history from which to base predictions, and user accuracy gives greater weight to these users. Finally, we note that the deep recurrent neural networks have virtually the same accuracy on all datasets, except for top-1 transfer accuracy. This shows that our deep recurrent neural network is not an overly complex model, yet still adds value when compared to a simpler model.

6.3 Future endpoint recommendation

We next study the effectiveness of our heuristics at predicting the endpoints a user will use in the future: more specif-

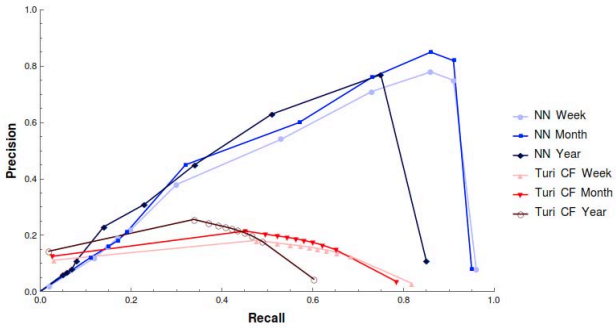


Figure 8: Transfer precision and recall for different recommendation thresholds

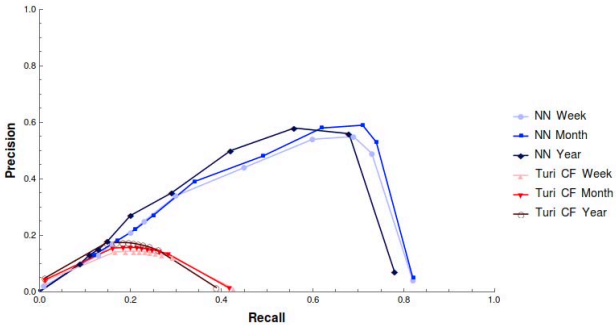


Figure 9: User precision and recall for different recommendation thresholds

ically, the endpoints that the user will use a week, month, and year after each transaction. We use the same heuristics and neural network combiner to predict these endpoints, but instead of recommending the top-1 or top-3 endpoints, we predict all endpoints that the neural network combiner weights above a certain threshold for the requested period of time. A lower threshold predicts more endpoints, increasing recall (the ratio of true predictions to endpoints actually used) but decreasing precision (the ratio of true predictions to all predictions). A higher threshold value, in contrast, recommends heavily weighted endpoints and thus gives a lower recall but higher precision. Precision and recall are both desirable in different situations; by adjusting our threshold value, we can observe obtainable precision vs. obtainable recall for each heuristic. As when predicting current transaction endpoints, we consider both total precision and recall (Figure 8) and user precision and recall (Figure 9). We show only the precision and recall of the Turi CF heuristic and neural network, as the other heuristics do not have obvious good endpoint weighting functions.

We see in Figures 8 and 9 that by adjusting the threshold the neural network maximizes precision and recall up to a point, whereas the CF approach performs significantly worse. Precision and recall for predictions over the next week and month are relatively similar, however, as expected, predicting endpoints for a year in the future is less accurate. A similar trend is observed for Turi CF. Our neural network is able to achieve approximately 75% transfer precision and recall for all cases, and 90% for the week and month intervals, significantly better than Turi CF. That is,

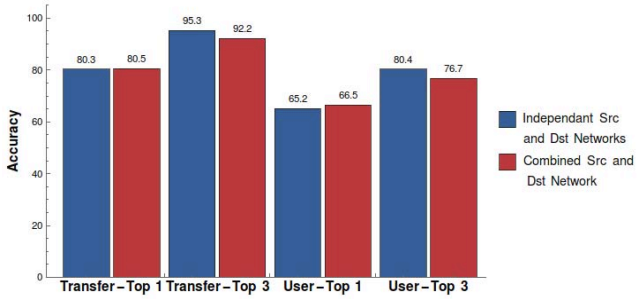


Figure 10: Independent source and destination networks vs. combined network

our model's predictions are correct at least 75% of the time, and if an endpoint will be used, our model will predict it at least 75% of the time. We again attribute Turi CF's significantly worse performance to its lack of domain-specific, content-based heuristics (e.g., owned endpoints, institution, and most unique users), which prevent it from producing a comprehensive list of the endpoints that a user could use, much less a list with few false positives. As in the previous section, our results are better when averaged over transfers rather than users because of the many users with few transactions.

6.4 Neural network analysis

We next explore what our neural networks have learned. While the neural networks perform well, it is not obvious what features are being used to predict usage. By exploring these features, we can gain insight into what factors are most indicative of future use.

First, we examine our choice to train two independent networks (for source and destination endpoints) rather than a single combined network for both (Figure 10). The combined network performs comparably to the separate networks for top-1 predictions (within 0.2% and 1.3%). However, individual networks perform better for top-3 predictions (3.1% and 3.7%). This shows there is a small benefit obtained by using two networks.

Next we investigate which pieces of information are most related to prediction accuracy. To do so, we simply remove individual features by setting its inputs to zero. If an important feature is removed, then accuracy will drop significantly. Figure 11 shows the results for both source and destination networks. Note the striking difference between the source and destination networks: the destination network is relatively unaffected by the removal of features (except the history heuristic), whereas the source network is affected by the removal of any feature. This tells us that the source and destination recommenders behave in very different ways: the destination recommender relies very heavily on a user's recent history when making recommendations; the source recommender, while still relying heavily on users' histories, makes significant use of other heuristics too. As these networks are quite accurate, this tells us that whether a user has recently used an endpoint is a very important factor in determining if that endpoint will be used as a destination, but many factors, including the endpoint's popularity at the user's institution and overall, are important in determining if the user will use that endpoint as a source.

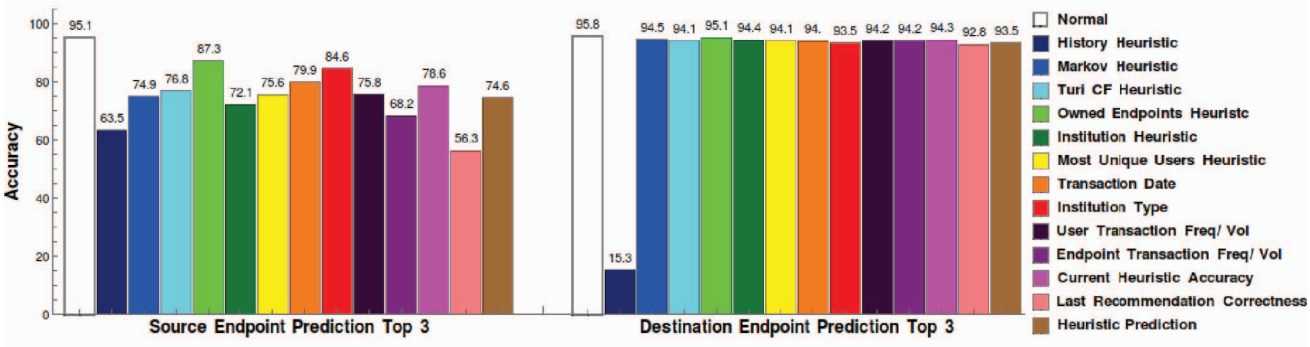


Figure 11: Top-3 next transfer accuracy with individual features removed

7. RELATED WORK

While many researchers have studied characteristics of scientific networks and methods for recommending data and analysis tasks in workflow systems, we are not aware of other work on recommending data storage locations.

Predicting transfer throughput is an important tool for optimizing transfer parameters and identifying and fixing software and hardware bottlenecks [23, 20, 18, 22]. In addition to creating throughput optimization schemes, recent studies of throughput prediction have studied the factors that affect transfer rate, including the effects of disk and raid controller bandwidth, data compression, and parallelism [15]; the effects of transfer protocol characteristics, such as buffer and window size [25]; and the detrimental effects of concurrent transfers and prioritization of transfers representing certain use cases [17]. Our work is differentiated both by its focus and also its approach, as none of these efforts use neural network methods for combining predictions. However, there is potential for these approaches to be mutually beneficial. For example, differentiating the throughput of transfers between the same pair of endpoints but initiated by different users requires some knowledge of those users, such as if they work in a field with relatively incompressible data; this knowledge could be used to predict which endpoints those users will use in the future. Furthermore, throughput prediction typically relies on various heuristics, whether average, median, or more complex statistical models. Not only does our neural network approach provide a powerful way of combining heuristics, but also a way to integrate problem information that is potentially relevant but difficult to use as a heuristic—something the previous ensemble methods used in throughput prediction do not. Finally, neural networks are able to learn complex and subtle relationships, and are reasonably amenable to analysis, allowing more insight into which factors influence various characteristics of scientific data transfers.

Another area of interest is the prediction of groups of files that are frequently transferred together. By identifying these groups, transfer performance can be improved using better caching and job scheduling algorithms [11]. While this problem is not directly related to endpoint prediction, we believe that many families of heuristics for characterizing transfers model both well. For example, the history heuristic is one of the best heuristics for predicting future endpoint usage, and historical data about which files were transferred together is often used to identify file groups. By using families of heuristics, such as those described in this paper, specifically the Markov model and neural network, we believe that larger and more strongly associated groups of files could be identified.

Workflow systems such as Galaxy, Kepler, and Taverna are used to orchestrate complex scientific analyses composed of independent applications. These systems provide interfaces for many users to develop and execute workflows, and therefore provide a rich source of information regarding users, workflows, applications and data [24, 10]. Following a similar motivation to our work, researchers have developed recommendation approaches to simplify usage of scientific workflow systems by recommending workflows, applications, and data to users [26, 5, 3, 9]. While these approaches are in a different domain, we have applied similar techniques to recommending endpoints. Furthermore, our work is complementary to these efforts as endpoint recommendations could be used to support the creation of workflows and selection of input data.

8. SUMMARY

We have developed and evaluated a collection of heuristics for recommending data locations to users. By leveraging rich sources of historical usage information and information about users, endpoints, and transfer settings we were able to create specialized heuristics that consider transfer history, user institutions, endpoint ownership and other information. While these heuristics performed well individually, the greatest performance is obtained by combining heuristics into an ensemble model using a recurrent neural network. Our neural network model was able to correctly predict the endpoints to be used in the next transfer with 80.3% and 95.3% accuracy for top-1 and top-3, respectively. It was also able to predict the endpoints that a user will use in the future with greater than 75% precision and recall. These results not only provide value for users but may also provide insights into how scientific big data is used and could be applied to develop better algorithms for accessing and transferring data.

We aim next to develop and deploy an online recommendation engine within Globus. We are also interested in characterizing, modeling, and improving the use of scientific big data by analyzing the performance of heuristics for different user profiles. Finally, we note that our deep recurrent neural network has virtually the same accuracy as a simpler neural network. While the simpler neural network still inte-

grates novel features and heuristics in a new way, we find the difference between the two models disappointing, and hope to improve our usage of deep recurrent neural networks in future work.

Acknowledgements

This work was supported in part by National Science Foundation grant NSF-1461260 (BigDataX REU) and Department of Energy contract DE-AC02-06CH11357.

9. REFERENCES

[1] Ranking factorization recommender. https://turi.com/products/create/docs/generated/graphlab.recommender.ranking_factorization_recommender.RankingFactorizationRecommender.html. Accessed Sept 2016.

[2] B. Allen, J. Bresnahan, L. Childers, I. Foster, G. Kandaswamy, R. Kettimuthu, J. Kordas, M. Link, S. Martin, K. Pickett, and S. Tuecke. Software as a service for data scientists. *Communications of the ACM*, 55(2):81–88, Feb. 2012.

[3] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. In *16th International Conference on Scientific and Statistical Database Management*, pages 423–424. IEEE, 2004.

[4] A. Ansari, S. Essegaier, and R. Kohli. Internet recommendation systems. *Journal of Marketing research*, 37(3):363–375, 2000.

[5] J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd. Gridflow: Workflow management for grid computing. In *3rd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 198–205. IEEE, 2003.

[6] K. Chard, M. Lidman, B. McCollam, J. Bryan, R. Ananthakrishnan, S. Tuecke, and I. Foster. Globus Nexus: A platform-as-a-service provider of research identity, profile, and group management. *Future Generation Computer Systems*, 56:571–583, 2016.

[7] K. Chard, J. Pruyne, B. Blaiszik, R. Ananthakrishnan, S. Tuecke, and I. Foster. Globus data publication as a service: Lowering barriers to reproducible science. In *11th IEEE International Conference on e-Science*, pages 401–410, Aug 2015.

[8] K. Chard, S. Tuecke, and I. Foster. Efficient and secure transfer, synchronization, and sharing of big data. *IEEE Cloud Computing*, 1(3):46–55, Sept 2014.

[9] D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor, and I. Wang. Programming scientific and distributed workflow with Triana services. *Concurrency and Computation: Practice and Experience*, 18(10):1021–1037, 2006.

[10] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540, 2009.

[11] S. Doraimani and A. Iamnitchi. File grouping for scientific data management: lessons from experimenting with real traces. In *17th international symposium on High performance distributed computing*, pages 153–164. ACM, 2008.

[12] I. Foster. Globus Online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing*, 15(3):70, 2011.

[13] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *14th International Conference on Artificial Intelligence and Statistics*, page 275, 2011.

[14] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[15] E.-S. Jung, R. Kettimuthu, and V. Vishwanath. Toward optimizing disk-to-disk transfer on 100g networks. In *IEEE International Conference on Advanced Networks and Telecommunications Systems*, pages 1–6. IEEE, 2013.

[16] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/ [Accessed, Sept 2016], 2015. Accessed Sept 2016.

[17] R. Kettimuthu, G. Vardoyan, G. Agrawal, P. Sadayappan, and I. Foster. An elegant sufficiency: load-aware differentiated scheduling of data transfers. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, page 46. ACM, 2015.

[18] C. Lee, H. Abe, T. Hirotsu, and K. Umemura. Predicting network throughput for grid applications on network virtualization areas. In *1st International Workshop on Network-aware Data Management*, pages 11–20, 2011.

[19] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727, 2012.

[20] M. Swany and R. Wolski. Multivariate resource performance forecasting in the Network Weather Service. In *ACM/IEEE Conference on Supercomputing*, 2002.

[21] S. Tuecke, R. Ananthakrishnan, K. Chard, M. Lidman, B. McCollam, and I. Foster. Globus Auth: A research identity and access management platform. In *12th IEEE International Conference on e-Science*, 2016.

[22] S. Vazhkudai and J. M. Schopf. Predicting sporadic grid data transfers. In *11th IEEE International Conference on High Performance Distributed Computing*, pages 188–196. IEEE, 2002.

[23] R. Wolski. Experiences with predicting resource performance on-line in computational grid settings. *SIGMETRICS Performance Evaluation Review*, 30(4):41–49, 2003.

[24] J. Yu and R. Buyya. A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing*, 3(3-4):171–200, 2005.

[25] D. Yun, C. Q. Wu, N. S. Rao, B. W. Settlemyer, J. Lothian, R. Kettimuthu, and V. Vishwanath. Profiling transport performance for big data transfer over dedicated channels. In *International Conference on Computing, Networking and Communications*, pages 858–862. IEEE, 2015.

[26] J. Zhang, W. Tan, J. Alexander, I. Foster, and R. Madduri. Recommend-as-you-go: A novel approach supporting services-oriented scientific workflow reuse. In *IEEE International Conference on Services Computing*, pages 48–55, July 2011.

[27] Z. Zheng, H. Ma, M. R. Lyu, and I. King. Wsrec: A collaborative filtering based web service recommender system. In *IEEE International Conference on Web Services (ICWS)*, pages 437–444, July 2009.

[28] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *4th International Conference on Algorithmic Aspects in Information and Management*, pages 337–348, 2008.