Feature Selection in Enterprise Analytics: A Demonstration using an R-based Data Analytics System

Pradap Konda[1], Arun Kumar[1], Christopher Ré[2], Vaishnavi Sashikanth[3]

[1]Department of Computer Sciences, University of Wisconsin-Madison
[2]Stanford University
[3]Advanced Analytics, Oracle

{pradap, arun}@cs.wisc.edu, {chrismre@cs.stanford.edu},
{vaishnavi.sashikanth}@oracle.com

ABSTRACT

Enterprise applications are analyzing ever larger amounts of data using advanced analytics techniques. Recent systems from Oracle, IBM, and SAP integrate R with a data processing system to support richer advanced analytics on large data. A key step in advanced analytics applications is *feature selection*, which is often an iterative process that involves statistical algorithms and data manipulations. From our conversations with data scientists and analysts at enterprise settings, we observe three key aspects about feature selection. First, feature selection is performed by many types of users, not just data scientists. Second, high performance is critical to perform feature selection processes on large data. Third, the provenance of the results and steps in feature selection processes needs to be tracked for purposes of transparency and auditability. Based on our discussions with data scientists and the literature on feature selection practice, we organize a set of operations for feature selection into the COLUMBUS framework. We prototype COLUMBUS as a library usable in the Oracle R Enterprise environment. In this demonstration, we use COLUMBUS to showcase how we can support various types of users of feature selection in one system. We then show how we optimize performance and manage the provenance of feature selection processes.

1. INTRODUCTION

Enterprise applications are using ever larger amounts of data to derive insights that can make businesses more agile and competitive. Advanced analytics – the application of statistical and machine learning techniques – enables enterprises to unlock the value of such large data to obtain better business insights. R has emerged as a popular environment for advanced analytics with a huge ecosystem of users contributing analytics techniques as open source R packages [1]. Major data management companies, including Oracle, IBM, and SAP have integrated R into data management platforms to enable enterprise users to leverage R-based analytics. Projects such as RIOT [6] have also attracted research attention to R-based analytics systems.

A key step in developing advanced analytics applications is *feature selection* – identifying important features of an entity or system being modeled [4, 5]. In colloquial terms, feature selection is about identifying the "key influencers" among other features in the dataset (a related problem is to select features from unstructured data such as text [2]). We now present an example business use-case of feature selection:

Example. *A telecommunication company provides voice and data services to its subscribers. It faces fierce competition from other telecom operators. Strategically, the company has decided to invest in analytics to find ways to retain its existing customers. Customers who switch to a competitor are called churned customers. The dataset contains features such as number of international calls, number of messages, data volume, traffic type, average number of service failures, gender, rate plan and thirty five other features, all of which have been collected over a period of three months. The task is to determine a subset of features (e.g., k out of n attributes in the dataset) as the key influencers of the target - propensity to churn in the next month. Based on the selected features, the company can enact better customer retention policies.*

Based on our conversations with data scientists and analysts at many enterprise settings – American Family Insurance, Deloitte, and customers of Oracle – we learned that feature selection is often an iterative, ad-hoc and a subjective process that is driven largely by a user's understanding of the entities under consideration. We also observed that feature selection is practised not just by data scientists, who have expertise in advanced analytics, but also by other kinds of users who have higher-level business requirements for the process. Based on our discussions with analysts, we categorize these users into three types based on what information they are interested in and the amount of control they exercise over the feature selection process:

- *Data scientists* are often involved in a hands-on and exploratory process for selecting features. They might use correlations among features to eliminate redundancies, transform features to reflect traffic trends,

Proceedings of the VLDB Endowment, Vol. 6, No. 12

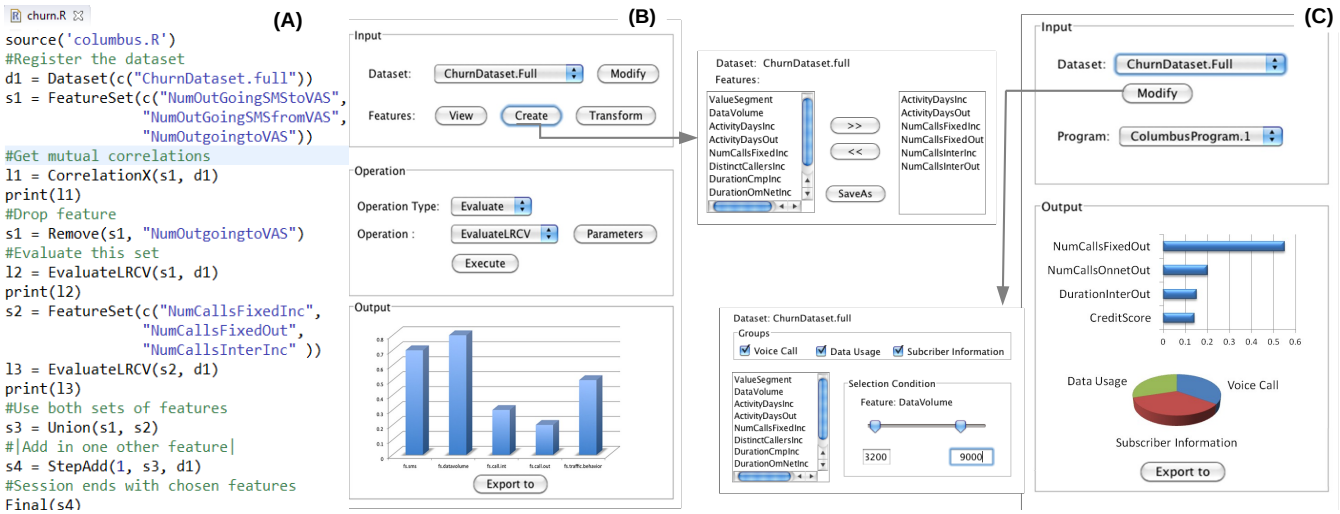

Figure 1: (A) A data scientist uses the R console to perform feature selection. (B) A managerial user uses a GUI to select operations, update parameters, and execute the operations. (C) A business analyst uses a GUI to select a dataset and a predefined COLUMBUS program to execute.

add features using step-wise addition, and finally get a subset of features that satisfy application requirements. They prefer command-line interfaces such as an R console and often build new algorithms and tools, which are then used by other users.

- *Managerial users* are often interested in higher-level overviews of the feature selection process and in using algorithms built by data scientists. They may classify features into groups such as traffic behavior and value segment, run feature selection algorithms, and combine the sets of features to get a final set of features. These users often use graphical user interfaces (GUI) that display algorithms and statistical tools to try multiple algorithms and change parameters.
- *Business analysts* are often interested in business decisions and use feature selection only to obtain important features. They use predefined off-the-shelf tools suggested by data scientists and are often not interested in technical details. Business analysts often use GUI to choose datasets and algorithms to run.

Although these different types of users have different needs and approaches to feature selection, they often describe feature selection in the same terms, using operations such as adding or dropping features, ranking features, or evaluating a set of features. Based on our discussions with analysts and the literature on feature selection practice [3], we organize a common set of feature selection operations in the COLUMBUS framework (see table 1) to provide a common stratum for all these users. We prototype COLUMBUS as a library usable in the Oracle R Enterprise (ORE) Environment. ORE enables users to analyze large datasets that are resident in the Oracle RDBMS using R language directly without having to write SQL. By building our framework on top of ORE, we can support these different types of users in the same system. We demonstrate how these three types of users – *data scientists*, *managerial users*, and *business analysts* – interact with COLUMBUS to perform feature selection for their tasks.

Increasingly, larger amounts of data are being brought in to analytics tasks such as feature selection to get more accurate insights. We demonstrate how we optimize the performance of feature selection operations in COLUMBUS to improve end-to-end runtimes.

Analysts often track information about the process of feature selection – known as *provenance* in the database literature to enable auditability and provide transparency. An analyst may want to query the intermediate steps and results of the processes for business purposes and to communicate the results for presentations. But different types of users are interested in different levels of detail. For example, a business analyst would only want to know if discriminatory features such as credit score are used, while a data scientist may want to know the intermediate results. We automatically capture provenance in COLUMBUS for all these types of users. We demonstrate a set of queries over the captured provenance and present the results graphically.

In summary, we demonstrate the following aspects of feature selection using COLUMBUS on real-world datasets:

- *Interface:* We demonstrate how three types of users – *data scientists*, *managerial* users, and *business analysts* – interact with COLUMBUS to perform feature selection.
- *Performance:* We demonstrate performance optimizations performed by COLUMBUS for the feature selection tasks of each user.
- *Provenance:* We demonstrate a set of queries over the provenance of the feature selection process and present the results graphically.

2. SYSTEM OVERVIEW

We now briefly describe the system architecture of COLUMBUS. There are five major components: Application Layer, Translator, Optimizer, Executor, and Provenance Manager. Analysts perform feature selection using a sequence of COLUMBUS operations, which constitute a COLUMBUS program.

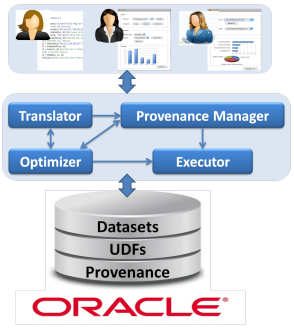

Figure 2: High-level architecture of our system to implement COLUMBUS. We show how each user interacts with the system to perform feature selection and pose queries over provenance.

Usage Modes. Similar to ORE, COLUMBUS operations can be used in two modes: *interactive* and *batch*. In the interactive mode, an analyst executes the operations one by one. In the batch mode, she provides the full COLUMBUS program to the system at once.

2.1 Front End

We provide custom user interfaces for the three different types of users, as illustrated in Figure 1.

- *Data scientists* are presented with an R console as shown in Figure 1(A). They can use COLUMBUS operations wrapped as R functions for analyses and also implement new algorithms, which can then be used by business or managerial users.
- *Managerial users* are presented with a GUI (Figure 1(B)) that contains options to view the dataset, create feature sets and add new features. They can run COLUMBUS operations by setting appropriate parameters and choosing feature sets.
- *Business analysts* are presented with a simple graphical user interface (Figure 1(C)) with options to select a dataset and choose a COLUMBUS program to run.

Our system tracks the provenance of the feature selection process. The users are presented with a GUI where they can pose a set of predefined queries over the provenance information and view the results graphically.

2.2 Back End

We now give a brief overview of how our system optimizes a COLUMBUS program. We observe two key properties of feature selection processes: (1) multiple operations often scan the same dataset independently, and (2) operations often access only a few features by projecting the dataset. Based on these two properties, we apply two classical database-style performance optimization ideas: *batching* of scan-based data accesses, and *materialization* of intermediate dataset projections. Once the program has been submitted, the Translator decomposes the COLUMBUS operations into workflows of atomic operations, each of which is a scan-based aggregation of the dataset. The Optimizer applies the batching and materialization optimizations, and the Executor runs the job on the data platform.

High-level Task	Example Operations
Descriptive statistics	Mean, Correlation
Train and score models	Linear and Logistic Regression, Cross-Validation
Automatic selection	Forward, Backward, Lasso
Semi-automatic selection	Stepwise Add and Drop
Manually choose features	Insert, Remove, Combine, Create new features
Data manipulations	Select, Project, Join

Table 1: Common high-level tasks and example operations in feature selection, based on Guyon and Elisseeff [3] and our interactions with analysts.

Performance Optimizations. We now briefly describe the batching and materialization optimizations applied by the COLUMBUS optimizer.

COLUMBUS operations often scan the same dataset independently. Batching can improve performance by sharing scans of a dataset both within and across operations. For example, a 10-fold cross-validation operation for logistic regression would perform 10 scans per iteration, if executed naively. Batching reduces the number of scans to 1. Also, COLUMBUS operations often access only a few features from the dataset. For example, an analyst might perform operations only on a set of 50 features out of 200 in a dataset. Using a materialized dataset with those 50 features projected might improve performance. But materialization presents a non-obvious tradeoff – the materialized projection is beneficial only if it is used often enough to pay off the time to materialize. We adopt a cost-based optimizer to make materialization decisions.

Provenance. In COLUMBUS, we capture the provenance of feature selection processes to enable querying of intermediate results. We explore two granularities of provenance for feature selection processes: (1) coarse granularity, in which we capture only the inputs and outputs of individual operations, and (2) fine granularity, in which we capture internal details of operations.

3. DEMONSTRATION DETAILS

We use the business case mentioned in section 1 and demonstrate how different types of users perform feature selection on the churn prediction task.

Data scientist. Data scientists often perform feature selection by being tightly involved in-the-loop.

Interface: A data scientist uses COLUMBUS operations wrapped as R functions to perform feature selection. She uses the R console interactively to construct sets of features and uses correlations to check for redundancies among features. She builds and evaluates models on the features, runs semi-automatic feature selection algorithms, and drops or adds features manually.

Performance: We present a graphical display that compares the runtimes of COLUMBUS operations with and without our performance optimizations.

Provenance: We show provenance queries posed by the data scientist: e.g., "Which feature sets were evaluated in a COLUMBUS program?" The result shows the evaluated

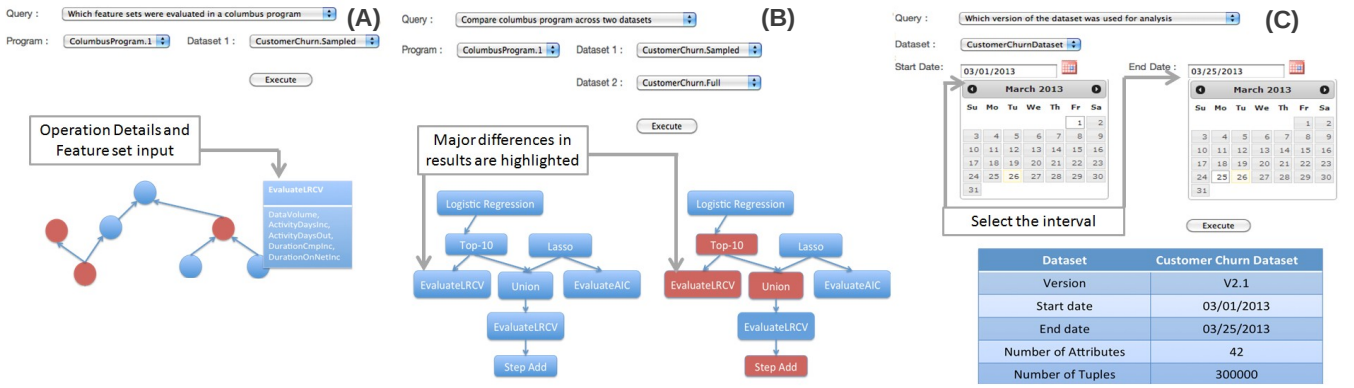

Figure 3: A web-based client to query provenance. The user selects a query from a drop-down menu, provides inputs and executes the query. Illustrating the results of a query posed by: (A) A data scientist – sets of features evaluated in a given COLUMBUS program are displayed. (B) A managerial user – results of all operations in a COLUMBUS program are compared across datasets. (C) A business analyst – details of the version of a dataset used for analyses in a given time interval.

feature sets and their containment relationships. The user can hover over the nodes to get more details (Figure 3(A)).

Managerial user. Managerial users often perform feature selection with GUI that provide statistical techniques and algorithms as COLUMBUS operations.

Interface: A managerial user uses a GUI to select feature sets using feature selection algorithms. She then constructs more feature sets using *create* (Figure 1(B)) and compares their cross-validation errors. She saves her operations as a COLUMBUS program and executes over a larger dataset.

Performance: We display the output of our optimizer on the input program. The VLDB audience can modify the optimization decisions interactively to see the estimated runtimes. The actual runtimes are then shown for a comparison.

Provenance: We show provenance queries posed by the managerial user: e.g., "How did the results of operations vary across the smaller and larger datasets?" The graphical result shows the operations marked with different colors to indicate major variation in the result. The user can hover over the operations to get more details (Figure 3(B)).

Business analyst. Business analysts often use off-the-shelf tools suggested by data scientists for feature selection.

Interface: A business analyst is presented with a GUI, where she applies filtering conditions in the dataset using *modify* (Figure 1(C)) to focus on high-value customers. She then selects a predefined COLUMBUS program to execute.

Performance: We show the COLUMBUS program invoked by the business analyst. We then display the runtimes with and without our performance optimizations.

Provenance: We show provenance queries posed by the business analyst on governance information, e.g., "Which version of the dataset was used for analyses?" The user selects the time interval and executes the query, she is returned with a table providing the version and other details about the dataset (Figure 3(C)).

In addition to the telecom dataset, the VLDB audience can use our system to perform feature selection on other real-world datasets such as an insurance-related dataset and a dataset from the U.S Census Bureau.

4. CONCLUSION

Feature selection is a key step in enterprise analytics applications. Based on our discussions with analysts in enterprise settings, we observe three key aspects of feature selection processes. First, different types of users, ranging from data scientists to business analysts, perform feature selection in enterprise settings. Second, performance is critical in feature selection processes, especially as larger datasets are used for analyses. Third, analysts want to query the provenance of feature selection processes in order to understand and communicate the results. We organized a common set of feature selection operations into the COLUMBUS framework and prototyped them on top of ORE. We use our system to demonstrate how different types of users perform feature selection, the performance optimizations applied by our system, and how the captured provenance can be used to answer queries about feature selection processes.

5. ACKNOWLEDGEMENTS

This research has been supported by the NSF CAREER award IIS-1054009, the ONR awards N000141210041 and N000141310129, the Sloan Research Fellowship and a gift from Oracle.

6. REFERENCES

[1] Project R. `r-project.org`.
[2] D. Antenucci, E. Li, S. Liu, B. Zhang, M. Cafarella, and C. Ré. Ringtail: A Generalized Nowcasting System. In *VLDB*, 2013.
[3] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, Mar. 2003.
[4] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature Extraction: Foundations and Applications*. New York: Springer-Verlag, 2001.
[5] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.
[6] Y. Zhang, W. Zhang, and J. Yang. I/O-Efficient Statistical Computing with RIOT. In *ICDE*, 2010.