Interactive View Recommendation with a Utility Function of a General Form

Xiaozhong Zhang
Dept. of Computer Science
University of Pittsburgh
xiaozhong@cs.pitt.edu

Xiaoyu Ge
Dept. of Computer Science
University of Pittsburgh
xiaoyu@cs.pitt.edu

Panos K. Chrysanthis
Dept. of Computer Science
University of Pittsburgh
panos@cs.pitt.edu

ABSTRACT

The utility function (UF) for ranking the views in traditional view recommendation (VR) approaches is defined a priori, thus failing to adapt to the analysis context, such as the current user and the analysis task. To address this shortcoming, we have proposed a new view recommendation paradigm, called *Interactive View Recommendation (IVR)*, and developed an IVR framework, coined *ViewSeeker*, which uses a novel active learning method to discover the UF. *ViewSeeker* effectively learns a UF of the linear form, which is assumed by most VR approaches. In this paper, we claim that the linear form is oversimplified and inaccurate, and UF should be of multiple functional forms (e.g., linear, non-linear, additive, multiplicative, etc.). We also propose a general *Utility Function Form (UFF)*, that fulfills commonly accepted UF properties. We then show that the commonly used learning methods cannot learn the general UFF well or easily, and propose specialized learning methods for learning certain special forms of the general UFF in an IVR setting.

ACM Reference Format:
Xiaozhong Zhang, Xiaoyu Ge, and Panos K. Chrysanthis. 2020. Interactive View Recommendation with a Utility Function of a General Form. In *Workshop on Human-In-the-Loop Data Analytics (HILDA'20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3398730.3399197

1 INTRODUCTION

Visualization recommendation tools are being used extensively nowadays to help users find interesting visualizations (i.e., views) from data. These tools usually employee a *utility function* (UF) to estimate the interestingness of the views and rank the views recommended to the users. A UF evaluates a view with respect to different *Utility Measures* (UMs), each representing an aspect of the view interestingness. Examples of UMs are *relevance* [6], *deviation* [11], *conciseness* [4], etc. The combined UM score produced by a UF for a view represents the degree of interestingness of the view. In traditional View Recommendation (VR) approaches [7, 8, 10, 11, 13], the UF is defined a priori. Thus, they cannot adapt to the analysis context, such as current user and analysis task. Some recent VR approaches [4, 5, 9] offer limited adaptability for a priori-selected UMs that form the UF.

In order to offer full adaptability, we have previously proposed a new VR paradigm, called *Interactive View Recommendation (IVR)*, which aims to dynamically discover the most suitable UF according to the analysis context during the analysis process [14, 15]. In addition, we have proposed a novel IVR framework *ViewSeeker* [15], which automatically helps the user in discovering the most suitable UF. *ViewSeeker* uses a novel active learning method that effectively minimizes the expensive user labeling effort during each IVR process by leveraging a linear form of UF (i.e., UMs are linearly combined in the UF) as prior knowledge.

Even though most VR approaches assume that the UF is of a linear form, we propose that the linear form is oversimplified and inaccurate, as the UFs in real-world analytic tasks are typically more complex. Below is an example to illustrate this point.

Assume that a doctor is trying to discover interesting views in clinical care data showing large differences between female and male patients with diabetes. In such a setting, a view showing small differences (i.e., low deviation) for the test outcomes relevant to diabetes (i.e., high relevance) will have low interestingness. Similarly, a view showing large differences (i.e., high deviation) for the test outcomes irrelevant to diabetes (i.e., low relevance) will also have low interestingness.

It can be seen from the above example that the UF could be in a form other than linear. Otherwise, at least one of the views in the above example should have medium or high interestingness. Motivated by this observation, in this paper, we propose to develop a more *general utility function form* (UFF) in order to cover a larger set of functions. Specifically,

HILDA'20, June 14–19, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-8022-5/20/06…$15.00
https://doi.org/10.1145/3398730.3399197

we make the following three major contributions in this paper:

- We identify commonly accepted properties of the UFs, and propose a general UFF that fulfills the identified properties (Section 3).
- We experimentally show in ViewSeeker that the general UFF cannot be learned well by commonly used learning methods in an IVR setting (Sections 4 & 5).
- We propose specialized learning methods for certain special forms of the general UFF, which can be used jointly for view recommendation in an IVR setting (Section 6).

In the next section, Section 2, we introduce IVR, which is the foundation of our work in this paper.

2 BACKGROUND

In this section, we present the necessary background details of our work. Specifically, we discussed view construction in the database context, traditional VR approaches, and interactive VR approaches.

2.1 Views & Data Visualization

In the context of structural databases. A view (i.e., histogram or bar chart) essentially represents an SQL query with a group-by clause over a database D [4, 11].

Under the typical multi-dimensional data models, data can be modeled as a set of measure attributes (i.e., numerical attributes) $M = \{m_1, m_2, m_3, ...\}$ and a set of dimension attributes (i.e., categorical attributes) $A = \{a_1, a_2, a_3, ...\}$. The measured attributes are a set of attributes that contain measurable value and can be aggregated. The dimension attributes are the set of attributes on which measured attributes are viewed. To formulate an SQL query with a group-by clause, we need to have a set of aggregation functions $F = \{f_1, f_2, f_3, ...\}$. Thus, each view v_i is the visualization of the query result of applying an aggregation function f on a measure attribute m, and grouping the aggregated values by a dimension attribute a.

Consequently, the View Space (VS), i.e., the total number of possible views is:

$$VS = |A| \times |M| \times |F| \tag{1}$$

Clearly, VS can be very large, especially for high-dimensional data.

2.2 Traditional View Recommendation

In order to recommend the set of k most interesting views from a large number of views, traditional VR approaches use UFs to rank the views. A UF maps a view to a real number indicating the interestingness of the view.

Definition 2.1. (*Traditional VR Problem*) Given a database D, a UF $u()$, and the size of the preferred VRs k, find the top-k views v_1, v_2, ..., v_k constructed from D that have the highest interestingness according to $u()$ among all possible views.

From the above definition, one can clearly see that in the traditional VR approaches [4, 5, 7–11, 13], the UF $u()$ is defined a priori, so that it will not be able to adapt to different analysis context such as the current user, the analysis task.

2.3 Interactive View Recommendation

To make the UFs adaptive to the analysis context, we have proposed *Interactive View Recommendation* (IVR) as a new VR paradigm [14, 15].

Definition 2.2. (*The IVR Problem*) Given a database D, a set of functions to calculate the UMs $U = \{u_1(), u_2(), ..., u_n()\}$, and the size of the preferred VRs k, find the top-k views v_1, v_2, ..., v_k constructed from D that have the highest interestingness according to $u()$ among all possible views, where $u()$ can be any combination of the functions in U, and is most suitable for the current analysis context.

In our IVR paradigm, the VR system interacts with the user during the analysis process to automatically learn the UF $u()$ that is most suitable for the current analysis context. An example is to ask the user to label example views with real numbers between 0 and 1 (with 0 being not interesting and 1 being most interesting), and use different learning methods to learn the UF based on user labels.

We call the functional form of the UF over the UMs, the *utility function form* (UFF), which defines the set of functions that can be used to combine the UMs. In most VR approaches, as well as In our previous work, the UFF is linear as shown below:

$$u() = \sum_{i=1}^{n} (\alpha_i \times u_i()) \tag{2}$$

where $u()$ is the UF, $u_i()$'s are the functions to calculate the individual UMs, and α_i's are non-negative real numbers representing the weights of the UMs with the sum of all α_i's being 1.

3 GENERAL UTILITY FUNCTION FORM

As discussed in the Introduction, the linear assumption of UFF is oversimplified, and fails to capture the complexity of UFs in real-world analytics. In this section, we develop a general UFF in three steps. Firstly, we identify the commonly accepted properties of UFs. Secondly, we analyze which properties can be fulfilled by a linear UFF, and which properties can not. Finally, we propose new UFFs that can fulfill the additional properties that are not fulfilled by a linear UFF.

Step 1 The UFs used in current VR approaches usually have the following properties: (i) A UF involves one or more UMs (P1). (ii) Each UM usually is a real number between 0 and 1 (P2). The higher the value, the more interesting the view is with respect to the UM. (iii) The final score from the UF is also a real number between 0 and 1 (P3). The higher the value, the more interesting the view is. (iv) The final score from the UF will increase if one involved UM increases, and the others remain the same (P4). In other words, the UF is monotonic with respect to each UM.

Step 2 It can be seen that the linear UFF in Equation 2 fulfills the above four properties. However, there are other properties of a UF that a linear UFF does not fulfill.

Step 3 The first additional property is that *dropping the score of certain UM to zero should bring the view interestingness to zero* (P5). It can be easily seen that this property cannot be fulfilled by a UFF in an additive form, because dropping to zero of any term in an additive function will not bring the function value to zero. In order to fulfill this property P5 capturing UM importance, we propose that a UF should be in a multiplicative form, as shown in Equation 3:

$$u() = \prod_{i=1}^{n} u_i() \tag{3}$$

It is worth noting that this multiplicative form also captures the UFF mentioned in [12] and used in [3].

The second additional property is that *not all UMs will bring the view interestingness to zero when their score drop to zero* (P6). In order to fulfill this property, we propose to group similar UMs together, and add an offset term to each group, as shown in Equation 4:

$$u() = \prod_{i=1}^{n} \left(\sum_{j=1}^{m} u_{ij}() + b_i \right) \tag{4}$$

where b_i's are non-negative real numbers representing the offset terms.

Note that we have omitted any non-essential slope/offset constants in Equation 4 for conciseness reasons. This rule applies to all the UFFs introduced in this paper.

It can be seen that the UMs are divided into n groups, so that the UMs in the same group evaluate the view interestingness from similar aspects. Each group involves m UMs and an offset term b_i. If a UM brings the view interestingness to zero when it drops to zero, then it will be in its own group (i.e., $m = 1$), and the offset of the group will be zero (i.e., $b_i = 0$). Otherwise, $m > 1$ or $b_i > 0$ or both.

The third additional property is that *the influence of each UM might not be constant over the domain of the UM* (P7). We define the *influence* of a UM to be the partial derivative of the view interestingness over the UM (i.e., $\partial u()/\partial u_{ij}()$). In order to fulfill this property, we propose to add exponents to the UMs, so that the resulting UFF will be able to express the influence changes, as shown in Equation 5:

$$u() = \prod_{i=1}^{n} \left(\sum_{j=1}^{m} u_{ij}()^{e_{ij}} + b_i \right) \tag{5}$$

where the e_{ij}'s are non-negative real numbers representing the exponents.

Since each UM is between 0 and 1, then a exponent e larger than 1 will be able to express the situation when the influence of the UM is smaller in the lower part of its domain and larger in the higher part of its domain. Similarly, an exponent e between 0 and 1 will be able to express the situation when the influence of the UM is larger in the lower part of its domain and smaller in the higher part of its domain.

Equation 5 is the general UFF that we propose in this paper. It not only fulfills the four initial properties of a UF (P1-P4), but also fulfills the three additional properties (P5-P7) that a linear UFF does not fulfill.

It should be clear that the general UFF (Equation 5) covers a much larger set of possible functions than the linear UFF (Equation 2). Actually, the linear form is a *special form* of the general form since the linear form can be derived from the general form by restricting the number of groups to one (i.e., $n = 1$), the offset in the group to zero (i.e., $b_1 = 0$), and the exponents for the UMs in the group to one (i.e., $e_{1j} = 1$).

As discussed below, our experimental evaluation of the performance of commonly used learning methods on the general UFF led us to propose two special forms of the general UFF in Section 6.

4 EXPERIMENTAL TESTBED

In this section, we will discuss the experimental testbed and the sets of experiments to be conducted.

Dataset We use the Diabetes dataset [1] in the experiments. The dataset contains 10 years (1999-2008) of clinical care data from 130 US hospitals and integrated delivery networks pertaining to patients with diabetes. It has 100,000 tuples, and 15 attributes, among which 7 are dimension attributes (i.e., categorical attributes) and 8 are measure attributes (i.e., numerical attributes).

Analysis Task We assume an analysis task of comparing the clinical care data between female and male patients. For view generation, we create a data subset for each gender, generate all possible views for each subset, and combine each view with the corresponding view from the opposite gender to form a comparison view (also called a view for simplicity reasons). With the remaining 6 dimension attributes (gender is used to split the data), the 8 measure attributes, and 5 aggregation functions, there are a total of 240 possible views.

Utility Measures We use three UMs as the representation for each view in the experiments. The first one is *deviation* as described in [11]. It measures the difference in the aggregate values between the two data subsets. The second one is *conciseness* as described in [4]. It measures the easiness of the view to be understood and remembered. The third one is *generality* as described in [6]. It measures how well the patterns in the view be generalized to the whole dataset. We define the generality of a view to being the ratio between the number of tuples with no missing measure attribute value and the number of tuples in the dataset. Since there is no missing data in the dataset, we have randomly assigned generality between 0.2 and 0.9 to the 8 measure attributes in the dataset.

IVR Workflow All experiments are run on the ViewSeeker platform [15]. Specifically, we set a ground truth UF, and let the system interact with the user to try to learn the UF. The system interacts with the user in an iterative fashion. In each iteration of the interaction, there are four steps: First, example views are randomly selected from all possible views. Second, labels for the example views are acquired from the simulated-user based on the ground truth UF. The label is in the form of a real number between 0 and 1, with 0 being not interesting at all, and 1 being the most interesting. Third, learners are trained from scratch on all example views with the UMs of a view as input features and the user label as the target label. Finally, all views are ranked based on the model prediction, and top-k views are recommended to the user. If the user is not satisfied with the recommendation, the system will go back to step one and start a new iteration.

Learning Methods Since the input features (i.e., UMs) and the user labels are all real numbers, we choose to use regression models as the learning methods [2]. We used eight regression models in the experiments. They are linear regressor, kernel ridge regressor, support vector regressor, k-nearest neighbor regressor, gaussian process regressor, decision tree regressor, gradient boosting regressor, and multi-layer perceptron regressor.

Evaluation Measure Assuming that the top-k views ranked by the ground truth UF is V^* and the top-k views ranked by the regression model are V, then the evaluation measure is the ratio between the cardinality of the intersected views between V^* and V, and the k, namely:

$$|V^* \cap V|/k \tag{6}$$

We call Equation 6 the *recommendation accuracy*, or accuracy for short. We measure the accuracy of the regression model with the highest accuracy after 10 example views have been labeled—we assume that 10 is an ideal number of examples that a user is willing to label in IVR. This rule of 10 example labeling applies to all experiments in this paper.

Table 1: Testbed Parameters

Total number of records	100, 000
Dimension attribute count	6
Measure attribute count	8
Aggregation function count	5
All possible view count	240
Individual utility measure count	3
Regression model count	8
Regression model hyperparameters	default values
Example view selection strategy	random
Example view count per iteration	1
Total iteration count	10
Performance measurement	Top-k accuracy
Accuracy model base	best performing model
Recommend view count (k)	5,10,15,20,25,30
Runs for each configuration	5

Setup Our experiments were performed on a Corei7 machine with 16GB of RAM memory. The platform is implemented in Python.

Experiments We conduct three sets of experiments. In the first set, the ground truth UF is in the general UFF (Equation 5). In the second set, the ground truth UF is in a special form of the general UFF, introduced in Section 6.1. In the third set, the ground truth UF is in another special form of the general UFF, introduced in Section 6.2.

The experimental parameters are summarized in Table 1.

5 GENERAL UFF EVALUATION

In the first set of experiments, we have tested six ground truth UFs in the general UFF. They are functions A1 to A6 in Table 2. The recommendation accuracy is shown in Figure 1 to Figure 6.

The average accuracies over the top-k's for the six functions are 58.6%, 78.3%, 85.7%, 92.2%, 80.0%, and 78.0%. The average accuracy over all functions is 78.8%.

This percentage is certainly not high accuracy. This indicates that the commonly used regression models cannot easily learn well the general form UFs with the default learner hyper-parameters and small amount of example views. Note that the accuracy increases to 95% with 120 example views. However, this is a high number of examples based on the assumed ideal/expected number of labeling in the IVR setting.

As a first step in addressing this challenge, in the next section we consider special forms of UFF that can be efficiently learned in IVR. As part of our future work, we will evaluate other learning methods capable of accurately and easily learning the general UFF.

Table 2: Ground Truth Utility Functions (D: Deviation, C: Conciseness, G: Generality)

#	Utility Function
A1	$D^{0.5} \times (C + 0.1) \times (G + 0.2)^2$
A2	$D^{0.5} \times (C + 0.2)^2 \times (G + 0.1)$
A3	$(D + 0.1) \times C^{0.5} \times (G + 0.2)^2$
A4	$(D + 0.1) \times (C + 0.2)^2 \times G^{0.5}$
A5	$(D + 0.2)^2 \times C^{0.5} \times (G + 0.1)$
A6	$(D + 0.2)^2 \times (C + 0.1) \times G^{0.5}$
B1	$D \times (C + 0.1) \times (G + 0.2)$
B2	$D \times (C + 0.2) \times (G + 0.1)$
B3	$(D + 0.1) \times C \times (G + 0.2)$
B4	$(D + 0.1) \times (C + 0.2) \times G$
B5	$(D + 0.2) \times C \times (G + 0.1)$
B6	$(D + 0.2) \times (C + 0.1) \times G$
C1	$D^{0.5} \times C \times G^2$
C2	$D^{0.5} \times C^2 \times G$
C3	$D \times C^{0.5} \times G^2$
C4	$D \times C^2 \times G^{0.5}$
C5	$D^2 \times C^{0.5} \times G$
C6	$D^2 \times C \times G^{0.5}$

6 SPECIAL FORMS OF GENERAL UFF

In this section, we introduce two special forms of the general UFF. These specialized learning methods can be jointly used in an IVR setting to provide view recommendations.

6.1 No Exponent UFF

The first special form we are going to introduce is called the *No Exponent* (NE) form:

$$u() = \prod_{i=1}^{n} \left(\sum_{j=1}^{m} u_{ij}() + b_i \right) \tag{7}$$

This form can be derived from the general form by having the exponent terms to be zero (i.e., $e_{ij} = 0$).

We have tested six ground truth UFs in the NE form. They are functions B1 to B6 in Table 2. The recommendation accuracy is shown in Figure 7 to Figure 12.

The baseline uses the original features, i.e., the UMs. The average accuracies over the top-k's for the six functions are 86.7%, 90.5%, 73.5%, 89.8%, 86.8%, and 83.4%. And the average accuracy over all functions is low, at 85.1%. The result for the baseline shows that the regression models cannot learn the NE form very well.

Therefore, we have developed a specialized learning method that extends the baseline with feature engineering. Specifically, we create new features that are multiplication of the original features (i.e., UMs). For example, if each view has three features A, B, and C, then we will create four additional features AB, AC, BC, and ABC for each view. The new features are created in the hope to catch the new variables generated by the NE form, to make the UF easier for the regression models to learn.

The experimental results have supported our hypothesis. The average accuracies over the top-k's for the six functions are 100%, 100%, 100%, 100%, 100%, and 99.4%. The average accuracy over all functions is 99.9%, which is satisfactory. The results show the effectiveness of our proposed method, which has an accuracy increase of 14.8% over the baseline.

6.2 Pure Multiplicative UFF

The second special form we are going to introduce is called the *Pure Multiplicative* (PM) form:

$$u() = \prod_{i=1}^{n} u_i()^{e_i} \tag{8}$$

This form can be derived from the general form by having each group involving one UM (i.e., $m_i = 1$) and having the offset terms to be zero (i.e., $b_i = 0$).

We have tested six ground truth UFs in the PM form. They are functions C1 to C6 in Table 2. The recommendation accuracy is shown in Figure 13 to Figure 18.

The baseline uses regression models directly. The average accuracies over the top-k's for the six functions are 49.1%, 39.1%, 83.7%, 33.8%, 94.9%, and 88.9%. And the average accuracy over all functions is very low, at 64.9%. The result for the baseline shows that the regression models cannot learn the PM form well.

Therefore, we have developed a simple yet effective method to help the regression models to learn the PM form accurately. Our method takes the logarithm of both the input features (i.e., UMs) and the label (i.e., view interestingness), before providing them to the regression models for training. Our method will also take a logarithm of the features before providing them to the regression models for prediction. The UFF after the logarithm operation of both sides of the PM form in Equation 8 is:

$$log(u()) = \sum_{i=1}^{n} e_i \times log(u_i()) \tag{9}$$

The intuition behind using the logarithm operation on the features and labels is that the logarithm operation will map the multiplicative form to additive form (Equation 9), which is much easier for the regression models to learn.

The experimental results have supported our hypothesis. The average accuracies over the top-k's for the six functions

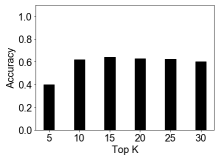

Figure 1: Accuracy for function A1.

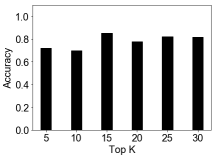

Figure 2: Accuracy for function A2

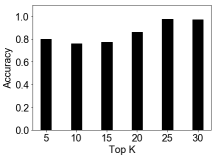

Figure 3: Accuracy for function A3

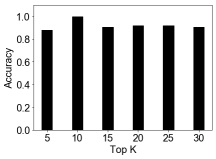

Figure 4: Accuracy for function A4

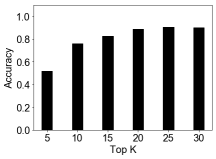

Figure 5: Accuracy for function A5

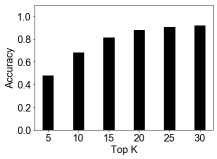

Figure 6: Accuracy for function A6

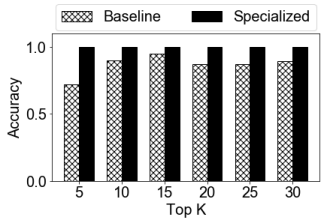

Figure 7: Accuracy for function B1.

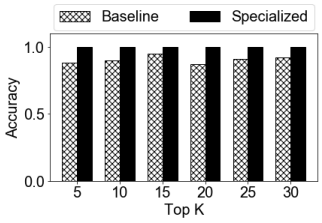

Figure 8: Accuracy for function B2.

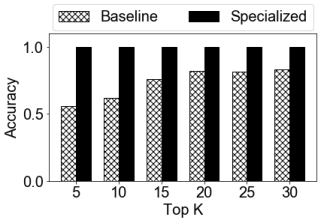

Figure 9: Accuracy for function B3.

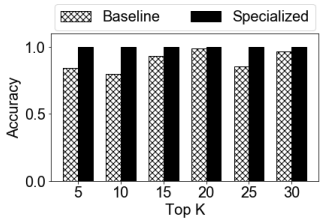

Figure 10: Accuracy for function B4.

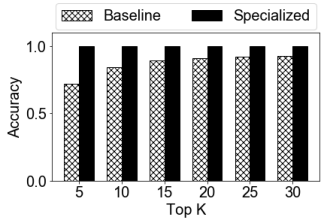

Figure 11: Accuracy for function B5.

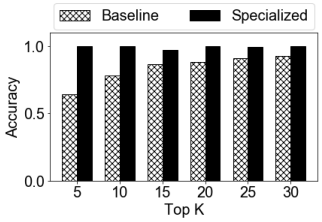

Figure 12: Accuracy for function B6.

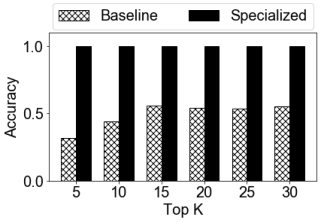

Figure 13: Accuracy comparison for function C1.

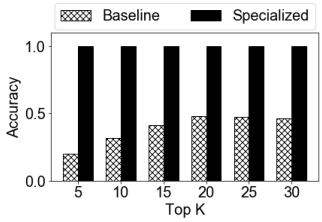

Figure 14: Accuracy comparison for function C2.

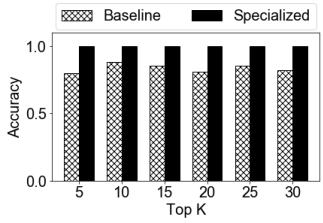

Figure 15: Accuracy comparison for function C3.

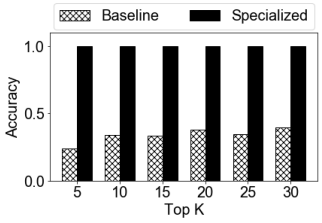

Figure 16: Accuracy comparison for function C4.

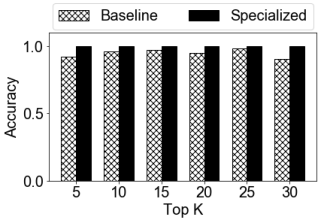

Figure 17: Accuracy comparison for function C5.

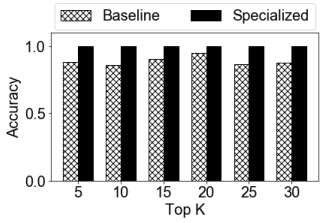

Figure 18: Accuracy comparison for function C6.

are all 100%, which is very satisfactory. The results demonstrate the effectiveness of our proposed method, which has an accuracy increase of 35.1% over the baseline.

7 CONCLUSION

In this paper, we first claim that the linear UFF assumption in most VR approaches is oversimplified and inaccurate. Then we analyze the commonly accepted properties of a UF, and propose a general UFF that fulfills all identified properties. Next, we experimentally show that the commonly used regression models cannot learn the general UFF well or easily. Finally, we propose specialized learning methods for certain special forms of the general UFF, which can be used jointly in an IVR setting for view recommendation.

Our next step is to conduct user studies to verify the effectiveness of the proposed general/special UFFs, and improve the learning methods for the general UFF in an IVR setting.

Acknowledgment: This work was partially supported by NIH award U01HL137159 and reflects the authors opinions.

REFERENCES

[1] 2019. Diabetes Data Set. https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008/

[2] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[3] Rui Ding, Shi Han, Yong Xu, Haidong Zhang, and Dongmei Zhang. 2019. QuickInsights: Quick and Automatic Discovery of Insights from Multi-Dimensional Data. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*. 317–332. https://doi.org/10.1145/3299869.3314037

[4] Humaira Ehsan, Mohamed A. Sharaf, and Panos K. Chrysanthis. 2016. MuVE: Efficient Multi-Objective View Recommendation for Visual Data Exploration. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*. 731–742. https://doi.org/10.1109/ICDE.2016.7498285

[5] Humaira Ehsan, Mohamed A. Sharaf, and Panos K. Chrysanthis. 2018. Efficient Recommendation of Aggregate Data Visualizations. *IEEE Trans. Knowl. Data Eng.* 30, 2 (2018), 263–277. https://doi.org/10.1109/TKDE.2017.2765634

[6] Liqiang Geng and Howard J. Hamilton. 2006. Interestingness measures for data mining: A survey. *ACM Comput. Surv.* 38, 3 (2006), 9. https://doi.org/10.1145/1132960.1132963

[7] David Koop, Carlos Eduardo Scheidegger, Steven P. Callahan, Juliana Freire, and Cláudio T. Silva. 2008. VisComplete: Automating Suggestions for Visualization Pipelines. *IEEE Trans. Vis. Comput. Graph.* 14, 6 (2008), 1691–1698. https://doi.org/10.1109/TVCG.2008.174

[8] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. DeepEye: Towards Automatic Data Visualization. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*. 101–112. https://doi.org/10.1109/ICDE.2018.00019

[9] Rischan Mafrur, Mohamed A. Sharaf, and Hina A. Khan. 2018. DiVE: Diversifying View Recommendation for Visual Data Exploration. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. 1123–1132. https://doi.org/10.1145/3269206.3271744

[10] Thibault Sellam and Martin L. Kersten. 2013. Meet Charles, big data query advisor. In *CIDR 2013, Sixth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 6-9, 2013, Online Proceedings*. http://cidrdb.org/cidr2013/Papers/CIDR13_Paper94.pdf

[11] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya G. Parameswaran, and Neoklis Polyzotis. 2015. SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. *PVLDB* 8, 13 (2015), 2182–2193. https://doi.org/10.14778/2831360.2831371

[12] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya G. Parameswaran, and Neoklis Polyzotis. 2015. SEEDB: Supporting Visual Analytics with Data-Driven Recommendations. https://data-people.cs.illinois.edu/seedb-tr.pdf

[13] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock D. Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (2016), 649–658. https://doi.org/10.1109/TVCG.2015.2467191

[14] Xiaozhong Zhang, Xiaoyu Ge, and Panos K. Chrysanthis. 2019. Leveraging Data-Analysis Session Logs for Efficient, Personalized, Interactive View Recommendation. In *5th IEEE International Conference on Collaboration and Internet Computing, CIC 2019, Los Angeles, CA, USA, December 12-14, 2019*. 110–119. https://doi.org/10.1109/CIC48465.2019.00022

[15] Xiaozhong Zhang, Xiaoyu Ge, Panos K. Chrysanthis, and Mohamed A. Sharaf. 2019. ViewSeeker: An Interactive View Recommendation Tool. In *Proceedings of the EDBT 2019 BigVis Workshop*.