Research Repository UCD

Title	Visualization for the masses : learning from the experts
Authors(s)	Freyne, Jill, Smyth, Barry
Publication date	2010-07
Publication information	Freyne, Jill, and Barry Smyth. "Visualization for the Masses : Learning from the Experts." Springer, 2010.
Conference details	18th International Conference on Case-based Reasoning (ICCBR 2010), 19-22 July 2010, Alessandria, Italy
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/2582
Publisher's version (DOI)	10.1007/978-3-642-14274-1_10

Downloaded 2023-04-07T10:52:29Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

Visualization for the Masses: Learning from the Experts

Jill Freyne¹ and Barry Smyth²

¹ Tasmanian ICT Center
CSIRO

GPO Box 1538, Hobart, 7001,
Tasmania, Australia

² CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin,
Dublin, Ireland

Abstract. Increasingly, in our everyday lives, we rely on our ability to access and understand complex information. Just as the search engine played a key role in helping people access relevant information, there is evidence that the next generation of information tools will provide users with a greater ability to analyse and make sense of large amounts of raw data. Visualization technologies are set to play an important role in this regard. However, the current generation of visualization tools are simply too complex for the typical user. In this paper we describe a novel application of case-based reasoning techniques to help users visualize complex datasets. We exploit an online visualization service, ManyEyes, and explore how case-based representation of datasets including simple features such as size and content types can produce recommendations to assist novice users in the selection of appropriate visualization types.

1 Introduction

So called “*knowledge workers*” are defined by the important role that information plays in their day to day work-lives. and over the last 10-15 years the Internet has provided them with a platform for information discovery, collaboration, and communication. Just as Web search engines have played a vital role when it comes to helping the typical user to access relevant information online, we are now witnessing the emergence of a new generation of information tools that will help users to make more sense of an ever-increasing quantity of raw data. Visualization technologies are set to play a key role in this regard, by helping users to better understand the relationships and messages that are often hidden within complex data.

Great strides have been made to bring a wide range of visualization options and tools to the masses. For example, Microsoft’s Excel offers 11 different types of chart (bar, line, pie etc.) and a total of 73 basic variations on these charts. Apple’s Numbers spreadsheet is similarly well equipped and even Google’s free Spreadsheets programme offers access to about 25 different variations of 6 different chart

types. No doubt readers familiar with the popular TED series (www.ted.com) will recognise the added-value that high quality visualization techniques can bring through the inspirational talks of Hans Rosling³. In short, visualization techniques can help us to make sense of complex data and by applying the right visualization technique to a given dataset can make all the difference when it comes to helping users to recognise the meaning and message of a given dataset [1–3].

Surely then all of this puts powerful visualization capabilities within reach of the average person? In truth, the answer is a negative one. The problem, of course, is that the average user is not a visualization expert and producing the right sort of visualization for a given dataset is far from trivial. How then can we help the average user to take advantage of what state-of-the-art visualization techniques have to offer? How can we provide meaningful assistance when it comes to helping a user to visualize their particular data set so that they too may access its hidden meaning? In this paper we describe a case-based recommender system that is designed to suggest visualizations to users, based on key properties of the dataset that they wish to visualize, by harnessing the past visualization experiences of other users.

Previous work in the area of visualization recommendation includes research into articulated task-orientated systems [4], early data-property based systems [5, 6], hybrid task and data based systems, which examine both user intent and the data at hand [7, 8]. More recent work aims to discover patterns in user behaviour in preparation of a dataset in order to predict visualization requirements [9]. Finally, closest to our work is work by Mackinlay from 2007 [10] where a rule based approach is taken to visualization recommendation. In that work the characteristics of a dataset including the structure and content determine the type of recommendation presented. This approach is similar to ours in terms of examining the dataset. However, rather than relying on the opinion of experts to determine the rules which are implemented in a non-flexible manner we believe that the ability to harness past visualization experiences can provide valuable insight into the visualization design process. This thinking suggests that there is an ideal opportunity to apply case-based reasoning methods to the visualization recommendation task. It provides opportunity for novice users to learn from the creativity of more experienced visualizers by suggesting imaginative ways to represent their data and also allows the system to increase the type and number of visualizations on offer without the need for additional rules to be added to a rule set.

The starting point for this work is a Web based “social” visualization platform called *ManyEyes* that was created by the Visual Communication Lab in IBM Research’s Collaborative User Experience group [11, 12]. In brief, ManyEyes is a web-based visualization platform that allows users to upload datasets, choose from a wide variety of visualizations, and make the results available to others. Each “visualization experience” encodes important visualization knowledge about the decisions taken by a user about how to visually represent a given

³ <http://www.gapminder.org/>

dataset. These decisions are not explicitly represented within the visualization structures but rather implicitly encoded according to the decisions taken by the user when producing a visualization. Thus, each visualization can be viewed as a concrete *case*, with features of the dataset providing the *case specification* and the resulting visualization configuration providing the *case solution*. In this paper we propose that these visualization cases can be reused to support the visualization of a new dataset, to make suggestions about appropriate visualization choices (e.g. visualization types). We describe a variety of recommendation strategies that explore a number of different ways to represent visualization cases plus a variety of metrics for assessing case similarity. We go on to evaluate these different approaches using real-world ManyEyes data to show how even relatively simple case-based reasoning techniques can inform useful recommendations.

In the next section we review the ManyEyes system, describing the features that are offered, and summarizing the visualization data that has been made available for this work. Section 3 describes the details of our recommender system, detailing a number of options for retrieving and reusing visualization cases. Then, in Section 4 we describe the results of recent evaluation, based on the live ManyEyes dataset, which demonstrate the potential for this recommender system to benefit ManyEyes users, especially casual or novice users. We conclude the paper with discussions of the analysis and an overview of future work.

2 ManyEyes: A Web-Based Visualization Service

ManyEyes (<http://manyeyes.alphaworks.ibm.com>) is an online browser based visualization tool designed not only to make sophisticated visualization easily accessible to web users, but also explore the potential for visualizations to serve as *social objects* that can be shared and that can serve as a focus of discussion and conversation. As such ManyEyes allows users to freely upload datasets and make them available to the wider public. It allows these datasets to be visualized using a variety of techniques (e.g. bar charts, histograms, line graphs, tag clouds, pie charts etc.); in total, ManyEyes supports 33 different visualization types. Users can try different visualization options on the same dataset and very often the same dataset might be visualized in a variety of different ways to reveal different relationships and concepts. For example, Figure 1 shows an example chart created from FDA survey data about the average lead content of vitamin supplements⁴.

The power of ManyEyes is certainly in the sheer range of visualization options that are available to end-users and the flexibility that is offered when it comes to experimenting with, and sharing, different datasets and visualization types. But this power also brings great challenges. ManyEyes is motivated by the desire to bring visualization to the masses but to the novice user choosing the right visualization for the right dataset can be a daunting task, and one that is not

⁴ <http://manyeyes.alphaworks.ibm.com/manyeyes/visualizations/fda-survey-data-on-lead-in-womens-an>

Fig. 1: An example of a ManyEyes histogram representing FDA survey data on fraction of lead contained in women’s and children’s vitamins.

well supported by ManyEyes, beyond the obvious trial-and-error process that it supports by default.

2.1 Data Upload

Raw data is uploaded to ManyEyes as freeform text or as tab-delimited files from popular applications such as Microsoft Excel or Open Office. During upload ManyEyes analyses the upload data to assign data types (e.g. text, numeric, etc.) to different dataset fields. In addition, users can add metadata to their datasets by adding a title, descriptive text, tags, and/or provenance information. In turn, ManyEyes also assigns information about the uploader’s user-id and upload date to each uploaded dataset.

2.2 Visualization Creation

ManyEyes supports 6 high-level visualization categories which cover a range of basic visualization functions to provide access to a total of 33 different visualization options. These basic categories include:

1. *Analysing Text* - Word tree, phrase net, and tag cloud type visualizations that are designed for the analysis of textual datasets.

2. *Comparing Sets of Values* - Visualization formats that are designed to provide a comparison of different sets of values, for example, bar charts, histograms, and bubble charts.
3. *Exploring the Relationships among Data Values* - Matrix charts, network diagrams, and scatter plots are used to help users explore the relationships between values.
4. *Visualizing the Parts of a Whole* - Pie charts and tree maps allow users to visualize the parts of a whole.
5. *Geographic Visualizations* - Map-based visualizations allow users to explore geographic datasets.
6. *Tracking Trends* - Line and Stack graphs are used for the visualization of trending data.

At visualization time, the user must select the visualization type that is most appropriate to their particular dataset and needs. While the designers of ManyEyes have provided some useful hints and tips about the type of visualization that might suit a particular dataset, this selection task is non-trivial, and it is one of the main barriers to entry for novice users. When a user selects a visualization type from the options provided, ManyEyes automatically generates the visualization, from the current dataset, automatically assigning chart parameters where possible, and asking for user conformation when multiple options exist.

2.3 Sharing & Discovery

One of the original motivations behind ManyEyes is the crowd-sourcing of different visualizations for a given dataset. It is expected that the right set of visualizations will help to *creatively explore* the dataset in a way that would simply not be possible in the context of a traditional solitary visualization scenario. To the creators of ManyEyes, visualization is a creative act, and datasets and their visualizations, are social objects that provide a focus for sharing and discussion. As such ManyEyes has been designed to support a variety of social interactions providing registered users with publicly accessible profile pages, for example, and supporting the active sharing, discussion and rating of datasets and their visualizations.

In the context of the “*knowledge worker*”, the availability of datasets and associated visualizations provides a rich environment from which non expert visualizers can learn. Novice or inexperienced users may discover datasets similar to theirs in order to decide how to effectively uncover the messages contained in their raw data. ManyEyes provides various methods for browsing and searching its repository of data and visualization pairs. We believe that case-based reasoning techniques could automate the process of discovering suitable visualizations for contributed datasets. By creating cases which represent simple dataset features such as the presence of numeric and textual content as well as the size of the dataset we aim to capture the expertise demonstrated by expert visualizers to assist users in selecting the best chart for their data.

2.4 The ManyEyes Dataset

In this work we are interested in supporting the ManyEyes user during the creative act of visualization itself, ultimately by suggesting useful visualizations to users at visualization time. As such we have created a case-based recommender system that harnesses the past visualization experiences of ManyEyes users as the basis for recommendation. We see this as a very appropriate and novel use of case-based reasoning. Visualization experiences are readily available, for example, and they implicitly encode complex visualization decisions that are often made by experienced and creative users. By selecting those experiences that are best suited for a given visualization task we can bring some of this creativity and experience to the novice user.

The dataset (case base) used for this work represents approximately 21 months of usage of ManyEyes from January 2007 and covers 33,656 separate dataset uploads, and 23,940 unique visualizations, from 15,888 registered users. It is worth highlighting that only 43% of uploaded datasets in this repository were successfully visualized; in other words more than half of the datasets do not have an associated visualization associated with it stored in the system, presumably, in part at least, because of the complexity of the selection task. In turn, about 40% of ManyEyes users who uploaded datasets never went on to create and store a visualization. This is surely a telling comment on the challenges faced by ManyEyes users when it comes to choosing and configuring suitable visualizations of their data. These are our target users: they are the novices who are motivated to upload data but for whatever reason did not successfully complete the visualization of their dataset.

In general there are two basic types of dataset in ManyEyes. *Text* datasets are *bag-of-word* type datasets whereas *tabular* datasets are the more traditional column-based datasets, using a mixture of data types. Here we focus on tabular datasets as they present the largest challenge for generating recommendations. The visualization of 9881 tabular datasets resulted in 16,848 different visualizations (1.7 per dataset).

3 A Case-Based Recommender for ManyEyes

The ManyEyes repository of datasets and visualizations is more than a simple collection of raw datasets and charts. It is a representation of visualization experiences, in the sense that each successful visualization represents a deliberate visualization attempt by a user, and is the product of a complex set of choices and decisions. Moreover we can reasonably assume, at least to some extent, that the user has made a good set of choices (considering information content and aesthetics, for instance) in terms of configuring an appropriate and useful visualization. Of course this assumption does not always hold up. For example, there is no doubt that many users create ineffectual visualizations. However, there are mechanisms available to evaluate the likely quality of a particular visualization effort. For example, ManyEyes allows users to rate visualizations. In addition, many datasets attract multiple visualizations and to the extent that different

users experiment similar visualization settings we might reasonably assume that a given visualization approach is a good one; for example if a particular dataset has been visualized 10 times and 8 of these use bar charts in various ways then we can assume that the bar chart is a reasonable chart type to apply to this dataset.

In short then, the dataset-visualization combinations in ManyEyes constitute a set of visualization *experiences* that can be represented as *cases*. To this end we propose to augment the existing ManyEyes system with a case-based recommendation component. When a new dataset is selected the case-based recommender first converts the dataset into a suitable set of features, uses these features to find a set of similar cases from the visualization case base, and then produces a ranked list of visualization types to suggest to the user. In the following sections we will summarize the case representation, retrieval, and ranking techniques that are used by this recommendation system.

3.1 Case Representation

We begin by assuming that each case represents a single dataset and a set of visualizations. Thus, each case, c_i is made up of a dataset component, d_i and a visualization component, $v_1, \dots, v_n$, as shown in Eq. 1. In fact there is also additional information that is sometimes available such as the rating associated with a particular visualization, r_i . In case-based reasoning parlance the dataset component corresponds to the *specification* part of a case, the visualization component corresponds to the *solution* part of a case, and the rating component can be viewed as the *outcome* of the solution. In this paper we will focus on the specification and solution side of visualizations cases, largely because the ManyEyes dataset is very sparse when it comes to the availability of ratings data.

$$c_i = \{d_i, v_1, \dots, v_n\} \quad (1)$$

In this work we are focusing on recommending a set of likely visualization types to a users when faced with a new dataset. Thus, the representation of the visualization component is relatively straightforward, each case solution is a set of visualization types used, $chart(v_1), \dots, chart(v_x)$, ordered by decreasing popularity in the visualizations, $v_1, \dots, v_n$, created from d_i . Going forward, one can envisage taking this a step further and reasoning about particular features of the visualization, such as the axis placement, label usage etc.

Each dataset is characterised by a set of simple features that relate to the type of data contained in the dataset. Each feature aims to represent the structure, content or metadata description of the dataset. For tabular datasets we extract *structural features* that include the number of textual columns, col_{txt} , the number of numeric columns, col_{num} , the number of data points (rows), $rows$. We compliment these by examining the *descriptive features*, a bag-of-words textual description derived from any metadata associated with the dataset, $desc$ (e.g., column headings, title etc.). In cases where tabular data contains numeric

columns we also extract *numeric features* that reflect data contained in the numerical columns such whether the column contains all positive, all negative or mixed values ($num_{pos}, num_{neg}, num_{mixed}$) such that num_{pos} is the number of numeric columns containing only positive values, num_{neg} is the number of numeric columns containing only negative values and num_{mixed} is the number of numeric columns with mixed values. In this way each case is represented as a feature-based dataset and solution as in Eq. 2.

$$c_i = \{col_{txt}, col_{num}, rows, desc, num_{pos}, num_{neg}, num_{mixed}, chart(v_i), \dots, chart(v_x)\} \quad (2)$$

3.2 Similarity and Retrieval

Given a new target case c_T (made up of a particular dataset and set of visualizations) the task of the recommender system is to locate a set of similar cases that can be used as possible visualization recommendations. We use traditional tried and tested similarity techniques using the case specifications above. To be specific, when computing the similarity between cases we use the similarity metric shown in Eq. 3 which calculates the relative difference between two cases by examining the differences between each feature $col_{txt}, col_{num}, rows, desc, num_{pos}, num_{neg}, num_{mixed}$.

The feature differences for all features are determined using Eq. 4 except for the *desc* feature which uses Eq. 5. In this instance uniform weighting is used.

$$sim(c_T, c_i) = 1 - \sum_{f \in \{features\}} w_f \bullet distance(c_T, c_i, f) \quad (3)$$

$$distance(c_T, c_i, f) = \frac{|c_T(f) - c_i(f)|}{max(c_T(f), c_i(f))} \quad (4)$$

$$distance(c_T, c_i, desc) = 1 - \frac{|c_T(desc) \cap c_i(desc)|}{max(c_T(desc), c_i(desc))} \quad (5)$$

In the evaluation section we will demonstrate that even these simple techniques work well when it comes to driving high quality recommendations, while at the same time leaving a number of options open for more sophisticated similarity techniques as part of future work. Thus, given a target case c_T we can use the above similarity techniques to produce a ranked list of n similar cases as the basis for recommendation.

3.3 Generating Recommendations

Each of the n cases retrieved will be associated with a set of visualizations. The same visualization type may occur in more than one case and so we can identify

a set of k different visualization types from these n cases. We need a way to rank these visualizations so that those that are associated with more similar cases are preferred over those that are associated with fewer, less similar cases. To achieve this Eq. 6 scores each of the n visualizations, v_i , as the sum of the similarity scores associated with the retrieved parent cases; $visualized(v_i, c_j) = 1$ if v_i is a chart type used to visualize c_j and 0 otherwise. The result is a ranked list of visualization recommendations, $v_1, \dots, v_k$ in descending order of their aggregate scores as per Eq. 6

$$score(v_i, c_T, c_1, \dots, c_n) = \sum_{\forall j=1 \dots n} sim(c_T, c_j) \bullet visualized(v_i, c_j) \quad (6)$$

4 Evaluation

This work is motivated by the fact that less than half (43%) of the datasets uploaded to ManyEyes have the resulting visualization saved to the system. We believe that a reasonable number of these “failed visualizations” were due, at least in part, to the confusion of choice that faced the novice first-time uploader while an additional proportion were simply not saved to the ManyEyes database but taken and used by their creators. In this work we implement sophisticated feature modelling and aimed to improve the visualization rate of ManyEyes by making proactive suggestions to the user about which visualization technique might best suit their dataset. In this section we will describe the results of a recent large-scale, off-line, leave-one-out style evaluation using a live ManyEyes dataset.

4.1 Set-up

The core ManyEyes test-set used in this evaluation covers a total of 14,582 unique datasets that have been uploaded by thousands of different users. These datasets have been variously visualized to produce 23,940 distinct visualizations which are represented as individual visualization cases. The diversity of visualizations created on ManyEyes is high with the most popular chart type being the *Bubble chart* which accounts for 17.2 % of the visualizations. The popularity of each chart type is detailed in Table 1

In an effort to eliminate low-quality visualizations from this core test-set we eliminated all those cases which were created by individuals who created only a single visualization. The motivation here is to remove contributions from very novice users as they probably do not reflect expertise in the area. This decision could have eliminated some high-quality visualizations but on average we expect the overall quality of the test-set to improve as a result. In the end the test-set of visualization cases used for this evaluation comprised of almost 10,000 tabular visualization cases.

For the purpose of evaluating our case-based reasoning approach to visualization recommendation we have developed 5 different recommendation strategies,

Bubble Chart 17.2%	Bar Chart 13.0%	Map 12.8%	Network Diagram 10.5%	Treemap 7.8%
Line Graph 6.5%	Scatterplot 6.3%	Matrix Chart 5.5%	Pie Chart 4.2%	Tag Cloud 4.2%
Stack Graph 3.1%	Stack Graph for Categories 3.1%	Block Histogram 2.3%	Treemap for Comparisons 1.8%	Word Tree 1.0%

Table 1: Popularity of Chart Types

2 of which represent simple benchmarks (*Random* and *Popular*) and the remaining 3 are different flavours of case-based reasoning that rely on different types of case specification data as the basis for similarity and retrieval. In summary, these strategies are as follows:

1. *Random* recommend a set of k random visualizations.
2. *Popular* recommend the k most popular visualizations in ManyEyes.
3. *Structure* this CBR strategy uses *structural features* only (such as the number of numeric and/or textual columns in a dataset) to produce a ranked-list of the top k visualizations from a set of n similar cases.
4. *Structure+Description* this more detailed CBR approach exploits both *structural* and *descriptive* (e.g. term-based information derived from dataset meta-data and/or column titles) features in its case representation to produce a ranked list of the top k visualizations from a set of n similar cases.
5. *Structure+Numeric Features* this alternative CBR approach exploits *structural* and *content* (e.g. information about individual dataset columns such as whether the contents were *positive*, *negative* or *mixed*) features to produce a ranked list of the top k visualizations from a set of n similar cases.

Our evaluation takes the form of a standard leave-one-out test. For each target case, c_T , we use its specification features (whether structural, description or content) as the basis of a new target dataset and generate a set of k visualizations using each of the 5 recommendation strategies above.

4.2 Measuring Recommendation Quality

There are many factors to consider, and different possible approaches to take, when evaluating the quality of a set of recommendations. In this study we look at two basic quality metrics. First, we consider the *accuracy* of the k recommendations produced by counting how frequently a given recommendation strategy produces a recommendation list that contains the *most* popular visualization for the dataset contained in the current target case. Many of these datasets will have been visualized in different ways, using different chart types for example. Our intuition is that if there is one visualization type that dominates then this is likely to be the best way to visualize that particular dataset and hence our focus

on looking for these most popular visualization types among our recommendation lists. So an *accuracy* of 60% means that the most popular visualization is present in 60% of the recommendation sets of size k . Of course there are many alternatives to estimating recommendation accuracy and, for what it is worth, we have considered alternative measures such as precision and recall across all visualizations, and the results produce are broadly in agreement with the result we will present for this particular measure of accuracy.

Fig. 2: Average Accuracy

Figure 2 clearly supports the use of a CBR recommendation strategy with all three CBR strategies significantly outperforming the two benchmarks (*Popular* and *Random*) across all values of k . It shows for example, that the *Structure* and *Structure + NumericFeatures* CBR approaches achieve 42% accuracy with their first recommendations in comparison to only 6% for both benchmarks. In other words, when we just focus on a single recommendation we find that the three CBR techniques suggest the most popular visualization chart almost half of the time, whereas the alternative strategies present this recommendation in less than 1 in 10 attempts. Indeed the *CBR* approaches reach an accuracy level of 70% when $k = 3$ whereas the benchmark techniques never significantly benefit from longer recommendation lists. We note that the three CBR techniques perform comparatively well with a very close coupling seen between the *Structure* and *Structural + NumericFeatures* algorithms but with the *Structure + Description* algorithm lagging slightly behind. This indicates that the information contained in the actual data which is being graphed is more reliable than the user specified associated metadata. Figure 3 shows the mean accuracy over all values of k for each strategy and again highlights the advantage

of the CBR techniques over the benchmarks and the variation between the three CBR techniques.

Fig. 3: Mean Average Accuracy

The second way we measure recommendation quality is to look for the position of these most popular visualizations among the recommendation lists: the lower the position, the better the recommendation list since the best visualization is appear nearer to the top of the list. Of course using a simple measure of *average position* benefits recommendation lists that do not contain the correct visualization. Thus as an alternative we actually calculate an *adjusted position* by assigning a $k + 1$ penalty to those lists that do not contain a correct (most popular) visualization. This is a conservative penalty because it assumes that the correct visualization is actually in position $k + 1$, which may not be, but it serves to at least remove some of the bias associated with a simpler measure of average position.

The results of the positional analysis of the recommendation techniques are presented in Fig.4 & 5. In terms of the average position statistic the CBR recommendation techniques are mixed in comparison to the benchmarks. For example, we see in Fig.4 the clear difference in performance of the CBR and benchmarks when the adjustment is made with all of the CBR techniques clearly superior to the benchmarks across all values of k . Once again we note the similar performance of each of the CBR approaches with only minor positional benefits observed when $k > 5$ for the *Structure + NumericFeatures* approach.

Fig.5 charts the mean adjusted positions for each approach across all values of k and clearly shows the superiority of the CBR approaches once again with the *Structure* and *Structure + NumericFeatures* approaches out performing the *Structural + Description* approach.

Fig. 4: Adjusted position

5 Conclusions

The objective of this work is to help users of a Web-based visualization system to produce better visualizations by harnessing visualizations that have been previously produced for datasets that are similar to their own. This has guided a case-based reasoning approach to recommendation where we view a dataset-visualization pair as a source of visualization knowledge. By reusing these visualization cases we can make high-quality recommendations to novice users.

We have described an evaluation that uses real-world visualization experiences to evaluate a range of different CBR approach, each of which explore the benefits of increasingly sophisticated case representations in order to drive recommendations. We compare these against two simple benchmarks. Overall the CBR approaches have been found to outperform the benchmarks, producing more accurate recommendation lists which contain appropriate visualizations nearer to the top of this list. Interestingly, we have noticed only minor additional benefits are gained from leveraging increasingly sophisticated case representations, at least in the test-set used in this study. Most likely this is a direct result of the large volume of visualization cases that are available. We note that no advantage is seen when the meta-data associated with a dataset is included in the case representation, this could be down to the small amount of data which people add to their datasets or the generic nature of the metadata. We note marginal increases in the performance of the CBR approach which examines the numeric content within each dataset. Examination of the data on across each visualization type did show that the *Structure + NumericFeatures* outperformed the simpler *Structure* approach for datasets which resulted in visualizations which fall under the category of *Tracking Trends* in particular when line

Fig. 5: Mean Adjusted Position

graphs were generated. This finding highlights the potential for more advanced case representation, possibly combining several other dimensions for comparison as highlighted by [10] such as whether the data is increasing or decreasing, the uniformity of the increase or decrease etc.

We have identified a number of areas where changes could affect the accuracy of our recommender.

1. *Noise and Novice users.* In this work we attempted to remove a certain amount of noise from the dataset by only including visualizations by creators who have created more than one visualization. We will look more closely at the users who have remained and their experience with visualization in order to generate an experience score which can be used to weight visualizations by their creators experience. This score could encompass data on the raw number of visualizations created or more intuitively the number of different visualization types a user has experience with.
2. *Ratings & Provenance.* ManyEyes maintains rating information and information about the creator of the particular visualization. In recent years there has been new work in the area of *provenance* [13] and *reputation* [14] that could be used to improve the recommendation algorithms by harnessing information about the source of a case and the reputation of the creator.
3. *Introducing Adaptation.* There is considerable scope for adaptation in this domain since recommending a visualization type is really just one part of a larger decision support problem. Users will benefit greatly from configuration support when it comes to actually using a particular visualization. This includes deciding which fields are associated with which axes, scale settings, etc. and these all provide opportunities for post-retrieval adaptation.
4. *Comparison to other Classification Techniques.* In this work we have compared our simple CBR technique to very simple alternative measures. With

work into creating sophisticated CBR representations planned we can also compare the performance of the CBR method with other classification approaches such as naive bayes, decision trees or neural networks. We also plan to compare our technique to a rule based algorithm [10] to determine its accuracy with its static counterpart.

References

1. Edward R. Tufte. *Beautiful evidence*. Graphics Press Cheshire, Conn., 2006.
2. Edward R. Tufte. *The cognitive style of PowerPoint*. Graphics Press Cheshire, Conn., 2004.
3. Edward R. Tufte. The visual display of quantitative information. *American Journal of Physics*, 53:1117, 1985.
4. Stephen M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Trans. Graph.*, 10(2):111–151, 1991.
5. Steven F. Roth, John Kolojchick, Joe Mattis, and Jade Goldstein. Interactive graphic design using automatic presentation knowledge. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 112–117, New York, NY, USA, 1994. ACM.
6. Jack Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986.
7. Michelle X. Zhou and Min Chen. Automated generation of graphic sketches by example. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 65–74. Morgan Kaufmann, 2003.
8. Elisabeth André and Thomas Rist. *The design of illustrated documents as a planning task*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1993.
9. David Gotz and Zhen Wen. Behavior-driven visualization recommendation. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 315–324, New York, NY, USA, 2009. ACM.
10. Jock. Mackinlay, Pat. Hanrahan, and Chris. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137, 2007.
11. Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. Manyeyes: A site for visualization at internet scale. In *IEEE Transactions on Visualization and Computer Graphics*, volume 13(6), pages 1121–1128, 2008.
12. Fernanda B. Viégas and Martin Wattenberg. TIMELINES Tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52, 2008.
13. David B. Leake and Matthew Whitehead. Case provenance: The value of remembering case sources. In *ICCBR*, pages 194–208, 2007.
14. Peter Briggs and Barry Smyth. Provenance, trust, and sharing in peer-to-peer case- based web search. In *ECCBR*, pages 89–103, 2008.