SpotLight: Visual Insight Recommendation

Camille Harris
Georgia Tech

Ryan A. Rossi
Adobe Research

Sana Malik
Adobe Research

Jane Hoffswell
Adobe Research

Fan Du
Adobe Research

Tak Yeon Lee
KAIST

Eunyee Koh
Adobe Research

Handong Zhao
Adobe Research

ABSTRACT

Visualization recommendation systems make understanding data more accessible to users of all skill levels by automatically generating visualizations for users to explore. However, most existing visualization recommendation systems focus on ranking all possible visualizations based on the attributes or encodings, which makes it difficult to find the most interesting or relevant insights. We therefore introduce a novel class of visualization recommendation systems that automatically rank and recommend both groups of related insights and the most important insights *within* each group. Our approach combines results across different learning-based methods to discover insights automatically and generalizes to a variety of attribute types (*e.g.*, categorical, numerical, and temporal), including non-trivial combinations of these attribute types. We then implemented a new insight-centric visualization recommendation system, SpotLight, which ranks annotated visualizations in visual insight groups. Finally, we conducted a user study which showed that users are able to quickly understand and find relevant insights in unfamiliar data.

KEYWORDS

Insight-centric visualization recommendation, data insight ranking

1 INTRODUCTION

Web-based visualization recommendation systems automatically suggest visualizations to support exploratory data analysis (EDA) and facilitate the visualization authoring process [3–7, 9]. These systems often leverage principles of expressiveness and effectiveness to produce recommendations that are agnostic to the specific data set. In this paper, we introduce a novel class of *insight-centric visualization recommendation systems* that automatically discover, rank, and visualize both specific insights and overarching insight-types (or classes of insights) in an arbitrary data set. Instead of displaying all visualizations together, they are grouped based on meaningful and intuitive insight-types and displayed to the user as rows of visualizations ordered by importance (see Figure 1). The key idea here is that the system scores and recommends the top insight-types, in addition to ranking the individual visualizations within each insight-type. This ranking enables users to quickly find the most interesting and relevant insights.

The main contributions of this work are as follows:

- **Insight-centric Visualization Recommendation.** We propose a novel class of insight-centric recommendation systems that enables users to quickly find insights in their data. We further contribute a new system of this type, SpotLight, which automatically discovers, scores, and ranks the insights as well as the insight-types that are most relevant in the data set of interest. We evaluated SpotLight with 12 users to demonstrate the efficacy, and show that users can quickly uncover a diverse and useful set of insights in an unfamiliar data set.
- **Insight-Type Ranking.** We introduce the notion of insight-type ranking and describe a general approach for recommending the most relevant insight-types for an arbitrary data set. The recommendation of insight-types (as opposed to just insights) improves the user experience making it significantly faster and easier for the user to identify the most important insights.
- **Insight Ranking.** We propose an insight recommendation approach that derives a score for each of the discovered insights within an insight-type. This ranking enables us to derive the top-k insight recommendations for each of the different insight-types. A key advantage of our proposed insight ranking approach is that it can naturally leverage multiple insight discovery methods, attribute combinations, attribute types, visualizations, and so on.

Figure 1: SpotLight is a web-based system for visual insight recommendation that automatically recommends visualizations for the most relevant insights in the data: (a) insights are automatically discovered; (b) insights are grouped into rows, scored, and ranked based on the *insight-type*; (c) these rows are then globally ranked and sorted; (d) users can interactively query recommendations through attribute filters; and (e) users can bookmark the most important insights.

2 PROBLEM FORMULATION

An insight is a property of the data that is unexpected, complex, deep, or relevant to the analyst, such as a strong linear correlation between two variables. Providing insights is arguably the main

WWW '23, May 1–5, 2023, Austin, TX, US
2023. ACM ISBN 978-1-4503-xxxx-x/18/06…$15.00
https://doi.org/10.1145/3366424.xxxxxxx

goal of information visualization [1]. However, none of the existing work on visualization recommendation focus on ranking the insight-types along with the visual insights of each type.

Definition 1 (Visual Insight Recommendation). *Let $\mathcal{I} = \{I_1, I_2, \ldots, I_{|\mathcal{I}|}\}$ denote the set of insight-types. Given an arbitrary data set $\mathbf{X}$, we define $\mathcal{F}_{I_i} = \{f_1, f_2, \ldots\}$ as the set of insight discovery methods for insight-type $I_i \in \mathcal{I}$ and $\rho_{I_i}(\mathbf{X}_1, \mathbf{X}_2, \ldots)$ as the set of insights found for that insight-type. An insight-centric visualization recommender system (i) automatically discovers the important insights for each insight-type $I_i \in \mathcal{I}$ using many different learning and statistical models $\mathcal{F}_{I_i}$ for many different attribute types and attribute type combinations; (ii) recommends the important top-r insight-types; (iii) scores and recommends the top-k insights $\rho_I(\mathbf{X}_1, \ldots, \mathbf{X}_k)$ for every insight-type $I_i \in \mathcal{I}$ and attribute type combination; and (iv) infers an appropriate visualization for each insight.*

Every insight-type $I_i \in \mathcal{I}$ has a set of attribute type combinations denoted as $\mathcal{A}_I$. For each attribute type combination $A \in \mathcal{A}_I$, we have a set of insight discovery methods $\mathcal{F}_I$ as shown in Table 1. For example, the "Two Variable Outliers" insight-type has two different attribute type combinations $\mathcal{A}_I = \{N{\times}N, C{\times}C\}$; each attribute type combination then supports several methods for detecting insights (*e.g.*, for $N \times N$, the supported methods include DBScan and IForest, among others). Furthermore, we also recommend many different visualizations for every insight type $I \in \mathcal{I}$, as shown in Table 1.

3 VISUAL INSIGHT RECOMMENDATION

We now describe our proposed approach for visual insight recommendation. See Table 1 for a summary of insight-types, attribute type combinations, chart-types, and methods supported.

3.1 Auto-Insight Discovery

The first step in the recommendation process is to automatically detect insights. To effectively identify insights for *any input data set*, we employ a variety of insight discovery methods. Given an arbitrary data set, each insight-type (and attribute combination) has many potential algorithms and the best algorithm depends on the data and its characteristics [8]. Since the "best" insight discovery method depends on the data and its characteristics, we instead use multiple insight discovery methods for each insight-type and attribute combination (as shown in Table 1) to ensure that our approach can find important insights, regardless of the user selected data set. In addition to the simple heuristic-based insight methods used in prior work (*e.g.*, IQR) [2], we use learning-based insight methods for many of the different insight-types as shown in Table 1. SpotLight also detects insights at a fine-granularity that are more meaningful and intuitive to the user. For instance, instead of considering a general insight-type called "outliers" as done in prior work [2], we distinguish outliers into meaningful subcategories, such as time-series outliers, multivariate outliers, single variable outliers, among others (Table 1). Furthermore, we consider multiple attribute combinations and many different attribute types for each insight-type shown in Table 1. The expressiveness of this approach contrasts with prior work [2], which only considers a single attribute combination and only supports numerical attributes.

In this work, we use many different methods to automatically discover insights for each insight-type. We provide a summary of these methods in Table 1. We categorize these methods into four main classes, including (i) **information theoretic**, (ii) **statistical**, (iii) **unsupervised learning**, and (iv) **supervised learning**. We adapt and leverage many of these methods in nonstandard ways for learning and automatically discovering insights from a wide range of different insight-types. To do this, we often have to first derive a different representation of the data, modify or change entirely the scoring function, etc. As such, we focus our discussion on the fundamentally different and novel ways these techniques were used in our insight recommendation framework. Importantly, the proposed insight recommendation framework is flexible as the insight discovery methods used for each insight-type and attribute type combination are completely interchangeable.

Table 1: We summarize the insight-types, and corresponding attribute type combinations and chart types supported for each insight. We also summarize the techniques used in the proposed framework for automatically detecting these insights, and color each one based on the approach used (information theoretic, statistical, supervised learning, unsupervised learning). N=numerical, C=categorical, T=time.

Insight Type	Attribute Type Combinations	Visualization Type	Methods
Timeseries Outliers	*N x T*	timeseries	autocorrelation • IForest • one-class SVM
	N x N x T	multiline timeseries	IForest • one-class SVM • LOF
	C x N x T	bar timeseries	temporal entropy • IForest • one-class SVM
Single Variable Outliers	*N*	boxplot	z-score • IQR • DBSCAN • LOF • one-class SVM
	C	bar	entropy
Two Variable Outliers	*N x N*	scatterplot	LOF • DBSCAN • k-means • IForest • one-class SVM
	C x C	stacked bar	PMI • ks-stat • IForest
Multivariate Outliers	*N x ... x N*	colored scatterplot	CBLOF • k-means • IForest • one-class SVM
	C x N x ... x N	stacked bar	CBLOF • IForest
Timeseries Correlation	*N x T*	timeseries	autocorrelation
Multivariate Timeseries Correlation	*C x N x T*	stacked bar timeseries	temporal entropy
	N x N x T	multiline timeseries	cross-correlation • cosine tdiff • Spearman tdiff
Two Variable Correlation	*N x N*	scatterplot	Pearson correlation • PCA • decision tree regression
	N x C	violin plot	logistic regression • decision tree classification
	C x C	stacked bar	Theil's U • Cramer's V
Nonlinear Correlation	*N x N*	scatter with LOESS	decision tree regression
	N x C	bar	decision tree classification • logistic regression
	C x C	stacked bar	Theil's U • Cramer's V
Rank Correlation	*N x N*	scatterplot	Spearman • Kendall • Goodman-Kruskal's γ
Timeseries Causality	*N x N*	multiline timeseries	Granger causality
Stability	*N x N*	timeseries	augmented Dickey-Fuller
Seasonality	*N*	timeseries	CH test
Spikes	*N*	timeseries	max local differences
Peaks	*N*	timeseries	k peaks of support
Irregularity	*N*	timeseries	augmented Dickey-Fuller
Skew	*C, N*	histogram	unbiased skew
Heavy Tails	*C, N*	histogram	kurtosis
Multimodality	*C, N*	histogram	bimodality coefficient
Dispursion	*C, N*	histogram	variance
Missing Values	*C, N*	pie	percent missing vaules
Joint Missing Values	*(C, N) x (C, N)*	pie	percent missing vaules per attribute pair

3.2 Insight Ranking

Once insights have been discovered, the next step is to create a ranking of insights to surface for the user. Previous work used a single heuristic for each insight-type and therefore did not require any scoring or ranking metric. In contrast, our approach automatically reveals insights of a specific type using multiple learning-based methods, which makes scoring and ranking the insights non-trivial. Ranking becomes even more complex given that the insights from an insight-type can leverage a different number of attributes of different types and combinations as shown in Table 1, which makes comparing and scoring individual insights difficult.

Given an insight-type $I \in \mathcal{I}$ and set of methods $\mathcal{F}_I$ for that insight type, the goal is to assign a score to each insight detected using $\mathcal{F}_I$. Suppose $p_I = |\mathcal{F}_I|$, then we may have p_I scores for every insight detected for insight type $I \in \mathcal{I}$. We define the utility function ϕ as:

$$\phi : \mathcal{X} \times \mathcal{F}_I \to \mathbb{R} \tag{1}$$

where $\mathcal{X}$ is the space of attribute combinations used by the methods $\mathcal{F}_I$ for insight-type $I \in \mathcal{I}$. We can derive a final score as follows:

$$\phi(\mathbf{X}_k, \mathcal{F}_I) = \frac{1}{Z} \sum_{f_i \in \mathcal{F}_I} \sum_{j=1}^{n} \big[g\big(f_i(\mathbf{X}_k, \Lambda_i)\big)\big]_j \tag{2}$$

where $\mathbf{X}_k$ is an arbitrary attribute matrix consisting of one or more attributes from $\mathbf{X} \in \mathbb{R}^{n \times m}$. In other words, $\mathbf{X}_k$ is an attribute combination matrix that may consist of one or more attributes. Further, $\big[g\big(f_i(\mathbf{X}_k)\big)\big]_j$ is the jth value from $\mathbf{s} = g\big(f_i(\mathbf{X}_k)\big) \in \mathbb{R}^n$. We define $g : \mathbb{R} \to [0, 1]$ and Λ_i as the set of hyperparameters for the learning-based insight method $f_i \in \mathcal{F}_I$. Suppose $f_i \in \mathcal{F}_I$ is one-class SVM, then Λ_i may include a kernel function $\mathbb{K}$ such as the non-linear RBF kernel or polynomial kernel along with other hyperparameters, such as γ or the degree of the polynomial kernel. In Eq. 2, $Z = |\mathcal{F}_I|n$.

Our formulation in Eq. 2 assumed that the output of each insight method was the same. However, in general, some insight methods may return scores for only the most relevant data points (as opposed to all n data points), or even a single score for the attribute combination. In this case, we can rewrite Eq. 2 as follows, where n_i denotes the number of scores returned by f_i:

$$\phi(\mathbf{X}_k, \mathcal{F}_I) = \frac{1}{|\mathcal{F}_I|} \sum_{f_i \in \mathcal{F}_I} \frac{1}{n_i} \sum_{j=1}^{n_i} \big[g\big(f_i(\mathbf{X}_k, \Lambda_i)\big)\big]_j \tag{3}$$

Eq. 3 assigns an insight score to the attribute combination $\mathbf{X}_k$ for insight type $I \in C$ using the methods $\mathcal{F}_I$. Now, we can obtain a ranking of the insights within the insight type $I \in C$ as follows:

$$\rho_I\big(\{\mathbf{X}_1, \ldots, \mathbf{X}_k, \ldots\}\big) \;=\; \underset{k}{\arg\text{sort}}\;\; \phi(\mathbf{X}_k, \mathcal{F}_I) \tag{4}$$

There are a number of novel aspects in the above formulation for ranking insights within an insight-type. More specifically, we are the first to (i) leverage learning-based insight methods, (ii) use more than a single method per insight, and (iii) propose and require an insight scoring and ranking function. Moreover, we describe a general mathematical framework for recommending visual insights.

Instead of using the normalized scores from every method to derive a final score for a given insight-type and attribute combination, we derive a final score for each data point (row of $\mathbf{X}_k$) based on the rankings given by each method. Using the ranks as opposed to the weights can help avoid biasing certain methods depending on the distribution of inferred weights. For a single j and attribute combination $\mathbf{X}_k$, we can generate a rank-based score as follows:

$$R_j(\mathbf{X}_k, \mathcal{F}_I) = \frac{1}{|\mathcal{F}_I|} \sum_{f_i \in \mathcal{F}_I} \pi_j\big(f_i(\mathbf{X}_k, \Lambda_i)\big) \tag{5}$$

where $\pi_j\big(f_i(\mathbf{X}_k, \Lambda_i)\big)$ is the position of the jth object in the ranking obtained from attribute combination $\mathbf{X}_k$ with method $f_i \in \mathcal{F}_I$. Therefore, $R_j(\mathbf{X}_k, \mathcal{F}_I)$ is the average rank of j across all methods and every method can be seen as having equal weight. Notice that Eq. 5 gives an overall ranking for each data point j (row in $\mathbf{X}$) whereas Eq. 4 provides a ranking of the overall attribute combinations $\{\mathbf{X}_1, \ldots, \mathbf{X}_k, \ldots\}$ across all n data points and $|\mathcal{F}_I|$ methods.

The rank-based score given by Eq. 5 can be used to appropriately annotate the visualizations for better visual insight recommendation. As an example, suppose we use the set of two variable outlier methods and use the methods as an ensemble to obtain an overall ranking of the data points by how much of an outlier each point appears to be. This ensemble ranking can be accomplished using the average rank of the data points given by the set of methods.

Note that if g is set to the min-max norm for each insight-type and attribute type combination, then by definition we are guaranteed to have a diverse ranking of visualizations for each insight-type. Intuitively, since min-max norm is applied to each attribute type combination independently, then one of the insights with that attribute type combination is guaranteed to score 1. Hence, if there are three attribute type combinations for a given insight-type, then the first three insights will be of different attribute type combinations.

3.3 Scoring and Ranking Insight Types

The insight-types are scored in a completely automatic and data-driven fashion. In particular, the ranking of the different insight-types (rows of insights and their visualizations) is driven by the amount of information captured by each insight-type across all the discovery methods. Let $Q_I = \{\mathbf{X}_1, \ldots, \mathbf{X}_k, \ldots\}$ denote the set of potential insights being scored for insight type $I \in \mathcal{I}$, then

$$\Psi(I) \;=\; \frac{1}{|Q_I|} \sum_{\mathbf{X}_k \in Q_I} \phi(\mathbf{X}_k, \mathcal{F}_I) \tag{6}$$

where $\Psi(I)$ is the overall score assigned to insight type $I \in \mathcal{I}$ for the dataset as a whole. As defined by Eq. 6, if an insight-type $I \in \mathcal{I}$ receives a relatively high score $\Psi(I)$, then there must be many important and highly weighted insights of that insight-type. The overall insight-type score is based on the insights discovered in the specific data set and their corresponding scores previously derived in Section 3.2. Using Eq. 6, we derive a global ranking of the insight types $\mathcal{I}$ by simply sorting them based on their overall scores:

$$\rho_{\mathcal{I}}\big(\{I_1, I_2, \ldots, I_{|\mathcal{I}|}\}\big) \;=\; \underset{I \in \mathcal{I}}{\arg\text{sort}}\;\; \Psi(I) \tag{7}$$

The insight-types are displayed to the user according to the ranking from Eq. 7. This ranking enables the user to quickly find the most relevant insights for the data set of interest. Furthermore, the overall insight-type ranking for a data set can be used to better understand the data quickly, *e.g.*, if the most important insight-types for a specific data set are all related to time-series (such as time-series outliers, time-series causality, and so on), then the user immediately knows that the temporal dimension in the data is important.

4 EVALUATION

To evaluate the effectiveness of the system, we conducted a user study of the system with twelve participants. The user study is designed to answer the following questions:

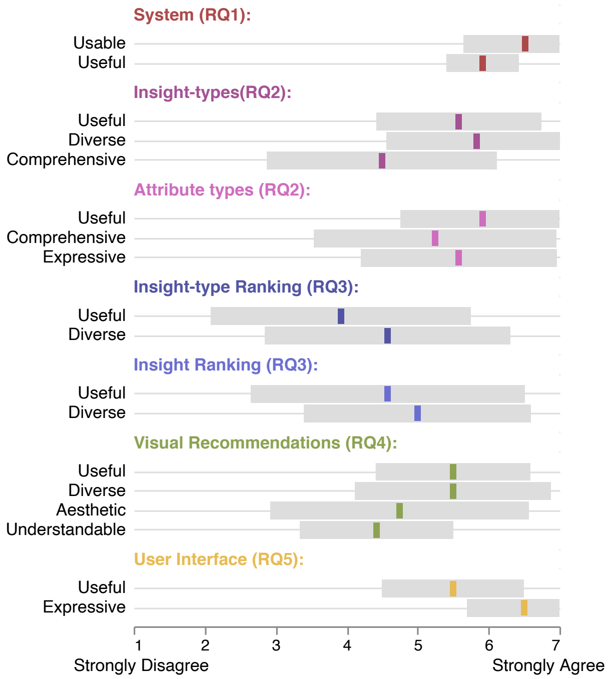

Figure 2: Overall ratings of SpotLight on a 7-point Likert scale, with one standard deviation shown in gray.

- **RQ1:** Does SpotLight help users generate hypotheses, find insights, and understand the data better?
- **RQ2:** Are the supported insight-types and attribute types useful, diverse, and comprehensive?
- **RQ3:** Are the recommended rankings (of both insight-types and insights) useful and do they provide diversity?
- **RQ4:** Are the visualizations used for the recommended insights useful, diverse, aesthetic, and easy-to-understand?
- **RQ5:** Does the UI allow users to find the most interesting insights and does it support different workflows for EDA?

Sessions were conducted by two researchers (one administered, one took notes). Participants were first trained on SpotLight's UI and features using a training data set. Participants were then asked to perform a free-form exploration of two data sets from the UCI ML Repository: `Weather` data and `Wholesale` customer profile data. No participant was familiar with either data set ahead of time and the order of the data sets was counterbalanced between participants to avoid fatigue and learning effects. During exploration, participants were encouraged to think-aloud and bookmark interesting charts. After exploring the data, participants were asked to verbally present their findings. Participants also completed a post-questionnaire with twenty-nine 7-point Likert questions and three questions about the usability and utility of SpotLight.

Overall results from the user study for answering RQ1-RQ5 are provided in Figure 2. Notably, participants felt the system is **useful** (**mean = 5.9**) and particularly **usable** for improving participants' understanding of the data (**mean = 6.3**). P5 noted that it was *"a good way to start my orientation in a dataset"*. P3 explained how the system provided *"immediate insights"* and could *"save a lot of time exploring the dataset to understand it"*. Participants were generally positive about the system and appreciated that it was comprehensive and automated when recommending insights. Even expert participants appreciated the ease-of-use, noting that *"I do a lot of this by hand using Python so I definitely can see value in some immediate analysis. I end up generating similar graphs but each takes time (and learning in Python)"* (P3) and *"I like that the visualizations are automated for me, so it's great to just bookmark and view them"* (P1). Most participants were able to find insights from the data based on the available insight-types, especially for insights that they would not have otherwise found. P2 noted that the system included insight-types that they would not think to use: *"I wouldn't think to correlate the numerical values with each other off the top of my head, but that makes sense"*. P7 similarly explained that *"a number of analyses listed were very useful in parsing out trends about the overall data"*. The visualizations were rated highly (**mean = 5.5**), especially with the inclusion of annotations, which *"help a lot to explain [the insights]"* (P2). Many participants commented about the **diversity of chart types**, citing this feature as a major strength of the system; P11 explained that *"The system provide more advanced visualizations that other tools provide"* and P7 noted that *"Some of the graphs were very visually appealing and indicated trends across different categories very clearly. Definitely worth using this tool for those graphs alone"*.

In addition, we also analyzed the rank of users' bookmarked insights as another means to understand the effectiveness of the insight and insight-type recommendations of our approach. Across all insight-types, 91% (weather) and 86% (customer wholesale) of the insights bookmarked by the users came from the top-10 insights recommended by SpotLight in their respective insight-type rows. This finding indicates that the ranking of insights (within each insight-type) from SpotLight were indeed useful. This observation is important, since there is clearly an exponential amount of visual insights that can be recommended for every different insight-type. In addition, users bookmarked multiple chart-types per insight-type, indicating that the diversity of charts was useful. We also found the insight-type ranking from SpotLight to be significantly correlated with the ground-truth ranking from the users in the study (with a `p-val`<0.01) using Kendall's τ rank correlation.

REFERENCES

[1] R. Chang, C. Ziemkiewicz, T. M. Green, and W. Ribarsky. Defining insight for visual analytics. *IEEE CG&A*, 29(2):14–17, 2009.
[2] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. *arXiv:1707.03877*, 2017.
[3] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *CG&A*, 39(5).
[4] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. Vizml: A machine learning approach to visualization recommendation. In *CHI*, page 1–12, 2019.
[5] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: towards automatic data visualization. In *ICDE*, pages 101–112, 2018.
[6] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *TVCG*, 25(1):438–448, 2018.
[7] X. Qian, R. A. Rossi, F. Du, S. Kim, E. Koh, S. Malik, T. Y. Lee, and J. Chan. Learning to recommend visualizations from data. *KDD*, 2021.
[8] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *TEVC*, 1(1):67–82, 1997.
[9] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *TVCG*, 22(1):649–658, 2015.