Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem statement

Often in modern multivariate analyses, data analysts rely solely on statistical estimators to explore the data. However, since each estimator inherently performs well or poorly under different settings, data analysts are unable to differentiate between the properties intrinsic to the dataset and the spurious properties the estimators added. Furthermore, the rapid increase in computing resources has led to the proliferation of high-dimensional datasets, which are more tedious to properly visualize. The problem is to find a way to visualize data to improve decision making and verify numerical models. To solve this, we develop a system that allow future analysts to visually confirm models determined through numerical methods. This enables data analysts to gain intuition on patterns in the data and recognize problems with various estimators. Furthermore, this tool increases standardization, and subsequently accountability, within data analysis. Our primary application focus is on graphical models and their financial applications. However, the visualization framework discussed in this work can be applied to any data analysis.

1.2 Motivation

Despite the fact that more than 2.5 quintillion bytes of data is produced daily to be analyzed, there is still a lack of accountability and consistency in modern data analyses. "Statistical thinking and methodology” has become the framework for disciplines such as education, agriculture, economics, biology, medicine, astronomy, geology, and physics [2], but it is unclear if data analysts agree on the procedures to analyze data. Little, a Professor of Biostatics at the University of Michigan, notes that “developing good statistical solutions to real applied problems, based on good science rather than ‘cookbookery,’ is far from easy” [5]. This lack of agreement and “cookbook” mentality of data analyses has far-reaching consequences. It is simple to run the data through a list of many estimators and cherry-pick the most “interesting” result. Similarly, an analyst can remove undesirable data points without justification or unknowingly fit egregiously incorrect models. Regardless of whether all these situations are performed maliciously or with good intentions, the art of data analysis is unclear without standards. The lack of clear-cut guidelines makes it difficult for analysts to discern the “truth” from the data and avoid the aforementioned pitfalls while simultaneously making it difficult for consumers of the resulting analyses to evaluate how trustworthy it is.

What is striking here is the lack of progress on this particular subject beyond the development of numerical methods. In computer science, a framework for “clean code” has been extensively documented and is the accepted industry standard for writing, interacting with, and thinking about code. But currently, in data analysis, analysts blindly depend on estimators and hypothesis tests to explore the data and have no justification of their analyses aside from asymptotic, mathematical guarantees. Hence, no such standards of “clean analysis” currently exist in data science despite the role its importance in financial decisions, judicial evidence, government policy, and scientific discovery. In this work, we tackle the problem by providing data analysts with a sophisticated visualization system to explore the data and numerical model in a different way. This way, future analysts can combine visual feedback with the numerical feedback from estimators to make better decisions during data analysis and provide clear justification of their decisions.

1.3 Solution overview

We focus on the specific problem of data visualization within the framework of "clean analysis." This is a specific element of the iterative, decision-making process of data analysis that provides guidelines and accountability for analysts. Specifically, we focus on developing a visualization tool to supplement in model selection after numerical methods have been applied. Model selection addresses the problem of determining which explanatory variables (commonly seen as columns of the matrix X) are informative to explain the variation in the response variable (commonly seen as the vector Y). This is related to the concept of "sparsity," a statistical term referring to the fact that many coefficients of a fitted model should be 0. Model selection with sparsity aids in the interpretability of the model since there are fewer variables for data analysts to understand. The visualization system aids in this process by checking the user's concept of an "interesting correlation" against the supplied numerical model. This allows the system to find visually interesting relationships that the numerical model may have missed and/or toss out relationships which turn out to be uninteresting.

While scatterplots and histograms are commonly used as a preliminary tool to explore datasets and verify the effectiveness of the fitted model, their effectiveness is lost on multivariate datasets with many variables. S. Liu et al. notes that "physical limitations of display devices and our [human's] visual system prevent the direct display and instantaneous recognition of structures with higher dimensions than two or three" [7]. This is troubling as high-dimensional datasets are found in numerous fields outside of finance. One solution is to manually plot each explanatory variable against the response variable, but this becomes computationally tedious and unfeasible to sort through when there are even a few hundred variables. This problem gets even more complicated when considering various transformations or combinations of explanatory variables ("interaction terms"). Due to its tediousness, some analysts may choose to utilize numerical methods alone, but they are left without a concrete way to verify their results. Although methods for dimension reduction have been developed [7], it is still unclear how the analyst can easily check the resulting model to ensure that the variables which were culled in the dimension reduction process are actually undesirable. Without considering these possibilities, important insight might be lost and the resulting model might be unsatisfactory unbeknownst to the analyst.

Regardless of whether high dimensional data visualization methods are computationally heavy or interaction heavy, user interactivity is still a vital component for processing high dimensions for visualization; it is simply a question of what degree [7]. We develop a system that first learns what visual patterns the data analyst finds promising, querying the user where the decision tree is ambiguous. It then automatically iterates through thousands of possible plots. Finally, it suggests relationships to exclude or include, which it believes to match the data analyst's interests, and their corresponding plots. This allows users to compare and contrast visual feedback with numerical algorithms for improved model selection.

1.4 Motivating example 1

With the univariate case (one observation variable x and one response variable y), we generate and alter a sample of 100 data points from a normal distribution. We have developed this data to illustrate an interesting phenomenon. Even in this simplistic setting, numerical tools commonly used in regression analysis can fail dramatically when not accompanied by corresponding visualization tools.

For the purpose of this example, imagine that a data analyst cannot produce any plots in the analysis, and the desired question to answer is if x contains explanatory power of y. The natural thing to try first is to fit simple linear regression and determine the significance of its coefficient. Doing this, the ANOVA table returns a significance level of 0.9109, extremely insignificant. When a constant-mean regression is performed, the resulting p value is equally as insignificant. The natural next step is to check the significance of each term in the linear regression model. Although the x-intercept $(0.460878, p < 2e-16)$ is significantly non-zero, the observed variable x is not $(0.008261, p = 0.911)$. Finally, a hypothesis test is used to determine whether or not the residuals are normally-distributed. The large p-value (0.5795) of the Shapiro test suggests it is. Next, one can check various correlations of x against the residuals or x against y, and each test yields no significant correlation. The raw data can be numerically printed out, but it's difficult to spot any trends in the data itself. Since there is little information that can gleaned on where to proceed next, it seems reasonable to conclude that the data is uncorrelated. The result is that the analyst scraps the regressions as they were "uninteresting."

Now assume that the analyst is allowed access to plotting tools in order to verify their numerical result. Once he/she plots the data (Figure 1.1), something peculiar happens. He/she finds that the analysis was completely incorrect. There is a strong v-shaped dependence between x and y. Nothing informed the analyst that there may be a nonlinear trend as none of the common hypothesis tests for linear regression failed, so the analyst would not suspect that the true trend was quadratic until the data was plotted. This example illustrates the fact that visualizations can lead to vastly different conclusions and act as a "sanity check" for the numerical results.

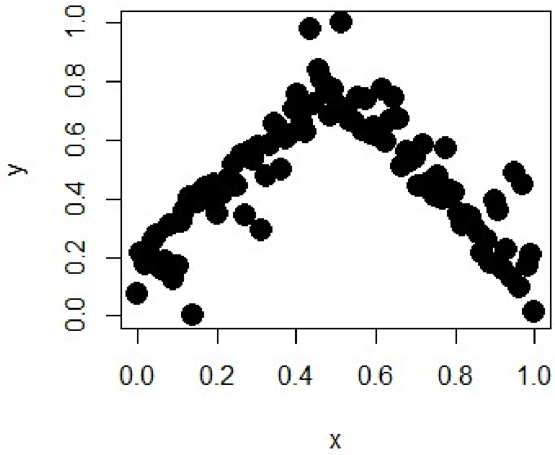


Figure 1.1: A plot of data that exhibits a strong quadratic trend but fails common tests of dependence. The code for this analysis may be found in Appendix A.1

1.5 Motivating example 2

In retrospect, it would have been simple to use polynomial regression or perform a non-linear dependency test in the previous motivating example. We argued that any unsuspecting analyst might have overlooked this since all the basic numerical methods were tried. Even if this is not true and the user did perform a non-linear correlation test, this does not necessitate that the results will be expected. We have developed another set of data to illustrate this point. Again, imagine that a data analyst cannot produce any plots in the analysis and is trying to determine if x contains explanatory power of y. Again, we fit a linear regression. The ANOVA table returns a significance level of 0.1896, which indicates that it is insignificant. The next step is to check the significance of each term in the linear regression model. Both the x-intercept $(-0.131, p = 0.488)$ and observed variable $x(-0.2699, p = 0.190)$ are not significantly different from zero. Finally, a hypothesis test is used to determine whether or not the residuals are normally-distributed. The large p-value (0.1632) of the Shapiro test suggests it is. Next, one can check the Pearson correlation of x against y (-0.189), but the p-value (0.1896) indicates that the two are uncorrelated (the correlation is not significantly different from zero). The final conclusion is that the data is uncorrelated. Again, the result is that the analyst scraps the regressions as they were "uninteresting."

Now assume that the data analyst is allowed to plot the data to verify their numerical solution. It is clear from Figure 1.2 that the data exhibit a negative linear trend. The outliers near the top left and top right were enough to drag the correlation coefficient towards to 0. It is difficult to notice these outliers by printing the raw data or only using numerical methods. And with a larger multivariate dataset, it would be even more difficult.

Notice that the analyst used a common test of linear correlation, and the data is clearly linear. The test, however, did not indicate that there was a significant relationship among the two variables; it was only after the data was plotted that this relationship was immediately noticed. Indeed, the data used in these examples was purposefully constructed to be dependent but bypass common tests for dependency. However, if it is possible to construct datasets in one-dimension that evade commonly-used numerical methods, it is believable that it is even easier to construct analogous datasets in higher dimensions.

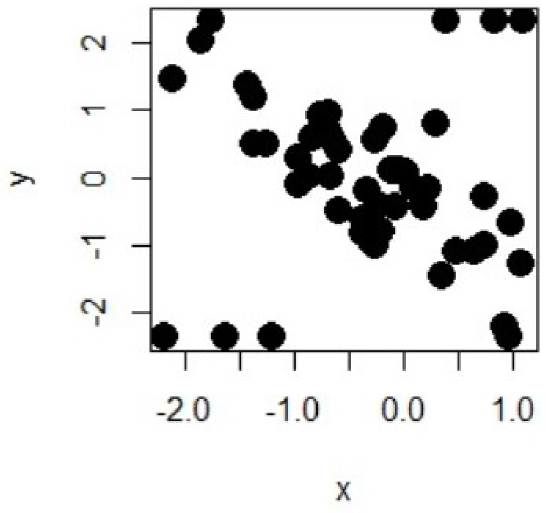


Figure 1.2: A plot of data that exhibits a negative linear trend but fails the Pearson correlation test. The code for this analysis may be found in Appendix A.2

1.6 Numerical data analysis

An important application of our system is in dependencies among financial equities. Determining the relationship among various stocks is especially useful when managing portfolios. One such consideration is a “buy and hold” tactic. This strategy is especially useful when transaction costs are high. Building a portfolio in such an environment means taking hedging into account so as to minimize the impact of bears on portfolio returns. Negatively correlated stocks hedge each other; as one goes down, the other goes up; in an ideal world, this results in zero net loss. The concept of sparsity is important once again. The analyst is looking to find negatively correlated instruments, but if the model is improperly selected, there may be too many connected variables. This makes interpretability and, ultimately, selection difficult as portfolio construction is performed with a fixed amount of capital.

In a world with low transaction costs, however, there is more to be gained (less to be lost, alternatively) by frequently rebalancing the portfolio rather than holding. A way to hedge the portfolio when rebalancing would be to select relatively independent assets; with perfect independence, rebalancing gains become a function of the volatility of the stocks rather than prices or other factors [6]. This is extremely important because putting in a trade for buy or sell an asset can move the market, leading to unexpected price swings. By selecting independent stocks, the analyst is able to minimize the price risk associated with rebalancing. Identification of independent stocks is another form of model selection; the goal is to select the proper model such that relationships (and in the case of stocks, the independence) among variables are easily seen. While a numerical model can certainly be fitted, it is still important to have a visualization tool to validate the results or risk losing millions of dollars.

Another financial application is the creation of a predictive model for a certain company's stock. This case is more in line with the definition of model selection noted earlier as the analyst is looking for dependence rather than independence. The analyst would like to filter out the independent assets X only to leave those which can explain the response in stock Y. This correlation can be positive or negative, and it can be a useful aid in trading as the predictive model, if it is to be believed, can signal price swings to come which opens up arbitrage opportunities. All of these methodologies can be boiled down to this one problem: determining the relationship among different stocks. This is not a trivial task, and there are a multitude of existing numerical approaches to quantify the relationships among different assets.

1.6.1 Existing methods

Equity market datasets are often large; when searching the space for correlated variables, we would like search through as many stocks as possible in order to be thorough and achieve the best possible portfolio. While there are many ways to quantify the relationship between different variables, we focus on correlation graphs and graphical models. A single correlation can be thought of as a regression of the response variable against only one observed variable; it is a "local" property because it compares the behavior of only two random variables. On the other hand, a single link in a graphical model can be thought of as a regression of the response variable against all variables in the space. It is more nuanced than that, but the idea is that a graphical model has a global property because it takes all of the other variables into account. Although it may seem simple to numerically quantify the dependencies with correlation as conditional dependencies are less intuitive to compute, correlations tend to fall short of the desired result due to the property of transitivity.

Correlation

Correlation graphs are one way to discover the dependency structure among the different stocks. Let $G = (V, E)$ be an undirected graph with vertices $V_1, ..., V_d \,\tilde{}P$ (a d-dimensional distribution) and edges $E_ij \in \{0, 1\}$. We set $E_{i,j} = 1$ when there is an edge between V_i and V_j, and 0 otherwise. An edge is drawn between V_i and V_j iff the two random variables are correlated. This graph can be drawn from a correlation matrix $\sum$ with the following heuristic:

If $\sum_{i,j} = corr(V_1, V_2) > p$, draw edge $E_{i,j}$ where p is the p-value for the desired confidence level

As mentioned earlier, we would like to find stocks that are as uncorrelated as possible in order to achieve the most robust portfolio. A common methodology to do this is to forego plotting entirely and numerically compute the correlation matrix then create a graph from that. By construction, the correlation matrix applies the same correlation computation to each set of observed and response variables. Correlation coefficients are rather limited and have their own flaws, so no coefficient is a one size fits all solution. For example, some variables could share a linear relationship with the response (ideal for a Pearson correlation) while others may not. While on the subject of the Pearson correlation, it should be noted that correlation graphs are still interesting because two variables that have a correlation coefficient near 0 may not necessarily be uncorrelated. A visualization tool can reveal this by showing outliers or patterns in the data that the analyst wasn't expecting (Section 1.5). Thus, the analyst needs to perform a visual check of the resulting correlation matrix rather than blinding accepting the results. This is important for improving accountability, as well, but it is a more difficult task than one might believe. In order to confirm that the coefficient used applies to each potential relationship, the analyst must plot all possible sets of data, which we have already established as computationally infeasible and tedious to sort through in high dimensions.

Furthermore, the result itself can be uninformative. Consider the property of transitivity, which states that if X is correlated to Y, and Y to Z, then X is also correlated to Z. Although correlation is not always transitive, situations where the correlation is close to 1 or 0, then the transitivity of correlation can be recovered and observed in the relevant data [8] Furthermore, correlation is a "local" property because it compares the behavior of only two random variables and ignores the rest of the data space, which may lead to an increasing amount of "false positives". Let us assume that the universe consists of Apple, Google, and Silicone (a manufacturer providing chips to both Apple and Google) stock. Suppose an analyst wants to model Google stock. Apple stock moves with Silicone stock as they depend on them for their chips. Similarly, Google stock (unbeknownst to the analyst) also moves with Silicone stock but not with Apple stock. The correlation between observed prices of Google and Apple stock will clearly and erroneously be positive without considering the way the stocks are connected to the other observed variable (Silicone stock). Given that correlation tends to be transitive, a correlation graph can have too many edges. This goes against the concept of "sparsity" and clutters the resulting space of observation variables that explain the response variable for the user. In the end, the analyst is left with an uninformative and unaccountable numerical solution.

Regression and graphical models

Graphical models alleviate some of the problems associated with correlation graphs, but they have their own set of problems, as well. Let G=(V,E) be an undirected graph with vertices $V_1, ..., V_d$ ˜P (a d-dimensional distribution) and edges $E_{i,j} \in \{0, 1\}$. We set $E_{i,j} = 1$ when there is an edge between V_i and V_j, and 0 otherwise. Do not draw an edge between V_i and V_j iff $V_i \perp V_j$ given V_k where $k \in \{1, ..., d\} \setminus \{i, j\}$. In other words, do not draw an edge if

$$P(V_i, V_j | V_k) = P(V_i | V_k) P(V_j | V_k)$$

This is known as a graphical model. The drawback is the difficulty in empirically computing conditional distributions and the problems associated with fitting distributions to real data. While there are simplifications that can be made for plotting (Section 2.1.3), the solution is not always so clear. However, the conditional independence of graphical models is more of a global property than correlation is because, for every pair of variables, it conditions on all the remaining variables. Returning to the universe of Google, Apple, and Silicone stocks, conditioning Google on Silicone and Apple on Silicone makes the relationship between the two clearly uncorrelated. Thus, although conditional independence tends to be more difficult to determine, it will tend to give a sparser network that is more interpretable for the analyst. The search through the rest of the data space is akin to the way regressions are fitted.

There are many ways to numerically perform model selection, and regression is the most common. In low-dimensional settings (where there are more samples than explanatory variables), the most common way is to fit a least-squares linear regression and perform a hypothesis test on each coefficient. The F-test is another useful tool for numerically informing the user if a regression model with more variables has significantly more explanatory power over a nested regression model. There are also ways to perform regression and model selection simultaneously. The most common estimator is the Lasso, the Least Absolute Shrinkage and Selection Operator (Tibshirani 1996). The drawback to all these methods is that their theoretical properties tend to either be asymptotic or reliant on assumptions. The former is unsatisfactory since datasets in practice are typically of a fixed size, far from the number of samples to achieve desirable properties analogous to when the number of samples goes to infinity. The latter is unsatisfactory because these assumptions are typically hard to verify or provide convincing justifications of.

Analysts may choose to use correlation over graphical models or vice versa as each has its own niche to fill. It has been shown that the rebalancing method, which leverages independent stocks and is more akin to graphical models, outperforms the "buy and hold" method, which leverages negatively correlated variables and naturally lends itself to the correlation approach [6]. Due to the nature of our financial application, we choose to utilize graphical models in our analysis of the data, but it is important to understand that both methodologies have the same need for a program that makes high dimensional visualization simple. This allows the analyst to make an informed decision on whether to confirm or reject the numerical result and improves decision-making in the financial industry.

1.6.2 Data collection

((I will explain how I collected the stock data (end-of-day historical prices for a thousand stocks over the last decade). I will also explain the data cleaning (to get rid of the dependency on time). This will be done by fitting a regression to the values and using the residuals as the new dataset. This makes it appear as if it is independent draws from a sample (perhaps this can be proved more rigorously).))

1.6.3 Analysis with existing tools

((Analogous to before, I will perform a full analysis without ever plotting the data. The goal is to find independent stocks for portfolio rebalancing. I will use a pre-made graphical model in order to fit the data numerically (as this is not the focus of the thesis this can be thought of as another step of data processing for the visualization system input). For simplicity (and since there are known, clear trends that we can expect of the conditional plots), I will probably use (and explain why I decided to use) the Gaussian graphical model.))

Chapter 2

Visualization System

It is extremely important to consider the way in which the system presents plots to the user as that can change the way the user perceives the data. Furthermore, user interactivity is a critical component of high dimensional analysis as noted earlier [7].

2.1 Scatterplot characterization

2.1.1 Characteristics of a "good" plot

The simplest scatterplot is the response against the observed variables. This, however, may not be the best way to ascertain independence for the user. This notion is illustrated in Figure 2.1. The left plot appears to be independent as its a cluster of points near the origin, but it's not entirely clear due to the multitude of stray points around the concentrated section. By looking at the outliers, it could also be argued that there is some dependency. However, applying the CDF in both directions creates a plot distributed on (0,1). It should be noted that this transformation is non-destructive and preserves dependency in the data if it exists. The data is clearly independent as the points appear to be uniformly distributed within the plot.

Restricting the plot to a unit box allows analyst's visual systems to focus on locations where there is low spatial frequency, which is ideal for detecting dependence [4]. The effects of this can be progressively observed by looking from the left to the right in Figure 2.2 below.

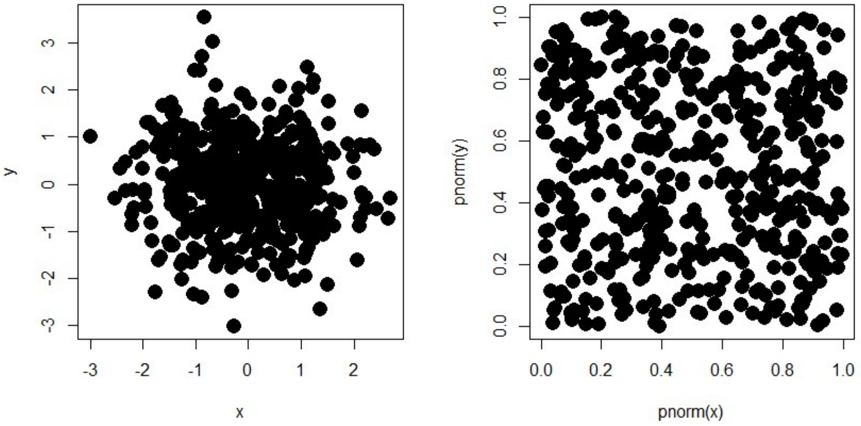
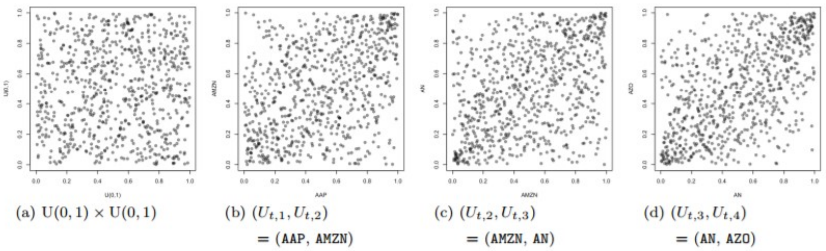


Figure 2.1: A plot of y against x with no transformation (left) and after the CDF is applied in both directions (right). The code for this example may be found in A.3



(a) U(0, 1) × U(0, 1)

(b) $(U_{t,1}, U_{t,2})$ = (AAP, AMZN)

(c) $(U_{t,2}, U_{t,3})$ = (AMZN, AN)

(d) $(U_{t,3}, U_{t,4})$ = (AN, AZO)

Figure 2 Scatterplots of (a) independent U(0, 1) random variables and (b, c, d) the pseudo-observation pairs $(U_{t,j}, U_{t,j+1})$, $j \in \{1, 2, 3\}$. Ticker symbol abbreviations: AAP = Advanced Auto Parts, AMZN = Amazon.com Inc., AN = AutoNation Inc., and AZO = AutoZone Inc.

Figure 2.2: Scatterplots of (a) independent $U(0,1)$ random variables and (b,c,d) the pseudo-observation pairs $(U_{t,j}, U_{t,j+1}), j \in \{1,2,3\}$. Ticker abbreviations: AAP = Advanced Auto Parts, AMZN = Amazon.com Inc, AN = AutoNation Inc., AZO = AutoZone Inc. Images from Hofert and Oldford 2016 [4]

2.1.2 Ordering

As people interact with graphs, they maintain a "mental map" of the graph. Federico and Miksch note that the importance of the mental map depends on various factors such as the user preferences and tasks that they must complete [3]. By "mental map," Federico and Miksch mean that when users label a new graph, they remember the previous plots that they labeled [3]. Supposing that the graphs that a program wants to show the user are already decided. Given a gradation of graphs (showing graphs that are most alike one after the other), users are less able to distinguish between differences than if they are shown graphs from different ends of the spectrum at different times [3]. While their work primarily concerns itself with ordering graphs for interpretability, their results can be applied to scatter plots. To put it concisely, the scatter plot display itself is not the only thing that matter. The ordering of the display matters, as well, and it is best to show the plots in an order that allows users to distinguish the differences among graphs that they have already seen. By improving their understanding of the plots, careful display ordering advances the accuracy of user responses. User responses can be thought of as observations of the users true preferences, and the ordering of plots as a way to future tune the precision of the decision tree (Section 2.3).

2.1.3 Conditional scatter plots

((There are many ways to produce this scatter plot. One way is to marginally plot one explanatory variable against the response variable. We experiment with another way where, similar to regression, we control for the behavior of all the other explanatory variables while making this scatter plot.))

2.2 Feature extraction from plot

2.2.1 Numerical features

((Our goal is to quantify various features of a scatter plot. One category of features are numerical features. These include Pearson correlation, tests of independence, mutual information criterion, etc))

2.2.2 Visual features

((The other category of feature are visual. How many points are near the center of the plot? How many points lie above the linear regression line?))

2.3 Active learning (Stage 1)

2.3.1 Decision tree classification of user interests

A decision tree is a method of classifying and labeling plots. An “active learner” adapts as the process moves forward, choosing its points of query intelligently. Active learning increases efficiency when searching through the hypothesis space, which is any fitted tree that agrees with the labeled data as much as possible [1]. Every time new data (a new label) is received, the hypothesis space shrinks as the label removes certain classifiers from the running [1]. An active learner queries from ambiguous parts of the current hypothesis space so as to shrink it as quickly as possible. It is problematic to start from scratch, however; how does the system determine the best first point of ambiguity when it knows nothing (the hypothesis space is everything)? While this problem is difficult, we can exploit the fact that the user is already providing a numerical model that they believe to be a good representation of the data which they would like the visualization system to check visually. Given this data, the system builds a decision tree that utilizes the various properties of the plots to determine whether one is interesting or uninteresting. Doing so greatly narrows the hypothesis space and makes it easier to determine points of ambiguity. However, to reconcile with the fact that the user wishes to check the numerical model and may not necessarily believe it is a good representation of fit, the learner must then perform several line-up tests (Section 2.3.4) to check whether the initial decision tree is a proper fit. As the user then proceeds to label various conditional plots as “interesting” or “not interesting,” the classifier learns the users interest and continues to evolve. This models plot characteristics that the user found interesting to study.

2.3.2 Rejection classification

So far, we have been discussing classification as black and white, interesting versus non-interesting. While interacting with the data visually, the user’s concept of what’s interesting in the specific dataset may evolve over time; at the beginning, they have no idea what the data looks like and where to set the bar for their own standards of dependence. There are several ways to take this into consideration.

First, the visualization system could assign a weight to the analyst's responses by trial number where the last few plots are more valuable than the first few. However, this may destabilize cases where the user's preferences dont end up changing, and it is difficult for the classifier to continuously rebalance each round given the weights (which, when applied to 2 black and white non-numerical responses, may also be vague). Secondly, the system can include an alternative option that allows the user to refuse to label a plot when it's too close to their decision boundary. This is welcome for the user who is not forced into making a decision he/she is uncertain about, but it is problematic for the learner as it causes the hypothesis space to remain unchanged rather than shrink. The point of the active learning segment is to have the user indicate to the learner which plots are interesting or not in order to let the computer better understand their preferences. By allowing for this option, the active learner may run for too long or return a poorly-defined tree. Finally, there is a way to consolidate the considerations within each methodology. The system can contain a third option that permits the user to "recycle" the plot. This allows to user to return to the plot later when he/she has learned more about what the data looks like and understand their own preferences better, and it ensures that the active learner will eventually receive data on the ambiguous plots that it has given the user to label. The main concern is that this could potentially de-balance the ordering procedure (Section 2.1.2), but the system can strategically insert the recycled plot between two plots it differs from with the constraint that the insertion location is after the current plot.

2.3.3 Selective plot generation

((To build a better classifier, we want to have the user label plots that the system (at the time) is uncertain about. This is where active learning comes in since we want to build our system to cleverly give users vital plots to label so the system can best learn the users interests. The system will have to determine which features it is uncertain about classifying and return a plot matching those characteristics to the user.))

2.3.4 Line-up tests

One of the pitfalls of data visualization is "apophenia," a phenomenon where the user sees patterns in random noise. Part of the reason for this is due to the vagueness of defining "independence" on a non-uniform domain and range (Section 2.1.1). Wickham et al. propose a line-up protocol that is similar to the Rorscach test where subjects are asked to interpret abstract blots of ink [9]. In the line-up test, users are asked to identify the real data from a set of n plots where n 1 plots are synthetically generated (Figure2.3). Analogously in the context of the classification problem, the learner generates 4 "uninteresting" plots (following the proposed decision tree) with 1 "interesting" plot and asks the user if he/she can identify the interesting plot. If the user is able to consistently identify the interesting plot, it is an indication that the current decision tree is a close fit of the users preferences.

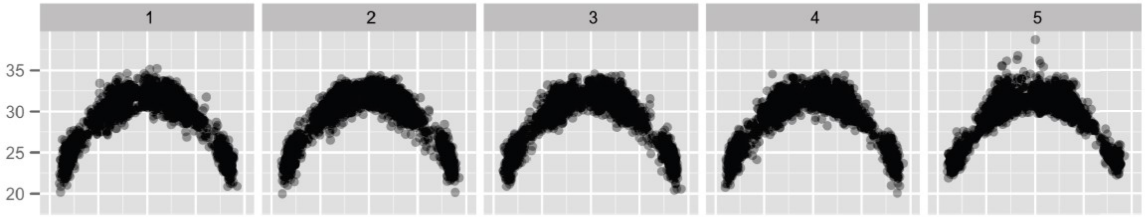


Figure 2.3: A line-up test for $n = 5$. Consistently identifying the raw data against the synthetic data indicates that the fitted model may not be good enough. Images from Wickham et al. 2010 [9]

2.4 Automated plot generation (Stage 2)

2.4.1 User interaction with active learning output

((Now that the system has learned the users interest, the user should be able to understand his/her own interests. We provide visualization tools of the resulting classifier itself such as a heat map, which there are also multiple ways to visualize))

2.4.2 Plot generation and feedback

((Equipped with the learned classifier of the users interest, the system can now automatically generate thousands of new plots and label them automatically. The most interesting plots are returned the user along with the explanatory variable (including possible interaction terms) corresponding to the plot))

2.5 Final system workflow

User provides numerical graphical model and data that they wish to analyze/confirm visually → program analyzes the plots among edges that do exist in the graphical model and extract relevant features (2.2) → initialize the decision tree given the resulting data (2.3.1) → perform a line-up test to determine how accurate the initialized tree is → selectively pick plots corresponding to ambiguous parts of the tree (2.3.3) → periodically generate line-up tests from ambiguous parts of the tree to confirm the current iteration of the decision tree is “correct” (2.3.4) → if the user passes a number of line-up tests, display the final classifier and heat map (2.4.1) → allow user to tweak if desired → output (2.4.2)

Chapter 3

Active learning

Literature review of AL methods, then AL simulations

Chapter 4

Graph comparison

Literature review of GC methods, then GC examples

Chapter 5

Results

Use the WHOLE visualizer system (with aforementioned inputs from ch 3 and 4) on the data

Compare results to numericala results

5.1 New analysis

((During the new analysis, I will make sure edges correspond with interesting conditional dependent plots & non-edges correspond to non-interesting conditional dependencies. I will also ask whether the fitted graph matches intuitively what we expect to see (For example, given a Gaussian GM, we expect to see linear relationships among nodes with edges)))

5.2 Results

((What new links were formed? Which links were deleted? Did we actually find the plots of those links interesting/not interesting?))

5.3 Implications

((Real-world context such as portfolio construction. Was my intuition correct? (For example, if Google and Facebook stocks were both in the dataset, I would expect a link between the two to be maintained)))

Chapter 6

Conclusion

Although this paper has focused on its financial applications, visualization systems nevertheless enable better decisions during any sort of data analysis. In the univariate case (Section 1.4, 1.5), it is clear that plotting has augmented the data analysis process by allowing the user to double-check their fitted models. In higher dimensions with more complex data sets and an inability to plot everything, it is even more believable that numerical methods cannot completely replace the valuable information obtained from visual methods. The visualization system developed in this text provides a way to safeguard data analysts against their own biases.

This visualization tool is one important step in streamlining the future of clean analysis. It provides a systematic way for confirming and suggesting dependencies among variables that match the analyst's concepts of a dependent (or "interesting" plot) and produces an explicit decision tree that allow others to understand and replicate the data analysis process. This removes the tediousness associated with high-dimensional data and allows the user to quickly see ways in which the numerical model may have fallen short of the "true" relationship between variables. In other words, it systematically provides the user a way to validate their methodology and model selection. Furthermore, it improves accountability in the analysis as it allows the decision-making process (in the form of a decision tree) to be clearer to those reviewing the results. Nevertheless, the graphical model that we develop is not fool proof, and further work can be done to refine the model and improve our concept of "clean analysis."

6.1 Further extensions

6.1.1 Estimator selection

((Estimator selection involves actively fitting the best model as opposed to “checking” a numerical model thats been given. This problem is more difficult to define, and the value that the visualization system adds is not as concrete))

6.1.2 Outlier removal

((Outliers are unavoidable in raw data and can skew results quite a bit. When can the system remove outliers? What criteria should it use?))

6.1.3 Edge-weighted graphs

((This is more of an extension on graphical models, but it is still related to estimator selection. In an edge-weighted graph, edges can be weighted depending on the type of conditional dependence negative or positive, assign 1 or -1. 0 (no edge) still implies conditional independence))

Appendix A

Implementation Details

A.1 Code for figure 1.1

```
#generate a reproducible dataset and scale to [0,1]
set.seed(10)
x <- seq(0, 1, length.out = 100)
y <- rnorm(100)
y <- (y-min(y))/(max(y)-min(y))

#sort the noise
y <- sort(y)
y <- y[c(seq(1,99,length.out=50), seq(100,2,length.out=50))]

#local swapping
for(i in 4:96){
y[(i-3):(i+3)] <- y[sample((i-3):(i+3))]
}

idx <- sample(1:100)
x <- x[idx]; y <- y[idx]

#####################
#numerical feedback

## fit linear regression
fitlm <- lm(y ~ x)
anova(fitlm)

## fit constant regression
fitconst <- lm(y ~ 1)
anova(fitlm, fitconst)
```

```
## see if any coefficients are significant
summary(fitlm)

## see if residuals are normally-distributed
shapiro.test(fitlm$residuals)

## various correlation tests
cor(x, fitlm$residuals)
cor(x, y, method = "pearson")
cor(x, y, method = "kendall")
cor(x, y, method = "spearman")

####################
#visual feedback
plot(x, y, pch = 16, cex = 2)
```

A.2 Code for figure 1.2

```
## generate the dataset
set.seed(10)
n <- 50
x <- sort(rnorm(n))
sd.vec <- c(seq(1, 1.5, length.out = 50), seq(1.5, 1, length
   .out = 50))
y <- -x + 0.5*rnorm(n, sd = sd.vec)
y <- scale(y)

y[c(1,5,10)] <- min(y)
y[c(n-10, n-5, n)] <- max(y)

#####################
#numerical feedback

## testing if regression is significantly different from
   fitting a constant regression
fitlm <- lm(y ~ x)
summary(fitlm)
anova(fitlm)

## test if residuals look like they came from a gaussian
shapiro.test(fitlm$residuals)

## correlation is not significantly different from zero
cor(x,y)
cor.test(x,y)

####################
```

```
#visual feedback
plot(x,y, pch = 16, cex = 3)
```

A.3 Code for figure 2.1

```
## generate the dataset
set.seed(10)
n <- 500
x <- rnorm(n)
y <- rnorm(n)

par(mfrow=c(1,2))
plot(x,y, pch = 16, cex = 2)
## apply the cdf
plot(pnorm(x),pnorm(y),pch = 16, cex = 2)
```

Bibliography

[1] Dasgupta, S. Two faces of active learning. `http://cseweb.ucsd.edu/~dasgupta/papers/twoface.pdf`, 2011.

[2] B. Efron. Why Isn't' Everyone a Bayesian? *The American Statistician*, pages 1–5, February 1986.

[3] P. Federico and W. Oldford. Evaluation of two interaction techniques for visualization of dynamic graphs. *arXiv preprint arXiv:1608.08936*, pages 1–15, August 2016.

[4] M. Hofert and W. Oldford. Visualizing Dependence in High-Dimensional Data: An Application to S&P 500 Constituent Data. *arXiv preprint arXiv:1609.09429*, pages 1–33, September 2016.

[5] R. J. Little. In Praise of Simplicity not Mathematistry! Ten Simple Powerful Ideas for the Statistical Scientist. *Journal of the American Statistical Association*, pages 359–369, July 2013.

[6] H. Liu, J. Mulvey, and T. Zhao. A semiparametric graphical modelling approach for large-scale equity selection. *Quantitative Finance*, pages 1053–1067, 2016.

[7] S. Liu, D. Maljovec, B. Wang, P. T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, pages 1249–1268, December 2016.

[8] Tao, T. When is correlation transitive? `https://terrytao.wordpress.com/2014/06/05/when-is-correlation-transitive/`, 2014.

[9] H. Wickham, D. Cook, H. Hofmann, and A. Buja. Graphical inference for infovis. *IEEE Transactions on Visualization and Computer Graphics*, pages 973–979, December 2010.