IPCA: An interactive system for PCA-based visual analytics

Article *in* Computer Graphics Forum · June 2009

DOI: 10.1111/j.1467-8659.2009.01475.x · Source: DBLP

Dong Hyun Jeong
University of the District of Columbia
48 PUBLICATIONS **530** CITATIONS
SEE PROFILE

Brian David Fisher
Simon Fraser University
116 PUBLICATIONS **1,326** CITATIONS
SEE PROFILE

Eurographics/ IEEE-VGTC Symposium on Visualization 2009
H.-C. Hege, I. Hotz, and T. Munzner
(Guest Editors)

Volume 28 (*2009*), *Number 3*

iPCA: An Interactive System for PCA-based Visual Analytics

Dong Hyun Jeong[1], Caroline Ziemkiewicz[1], Brian Fisher[2], William Ribarsky[1], and Remco Chang[1]

[1]Charlotte Visualization Center & UNC Charlotte, USA
[2]School of Interactive Arts+Technology& Simon Fraser University, USA

Abstract

Principle Component Analysis (PCA) is a widely used mathematical technique in many fields for factor and trend analysis, dimension reduction, etc. However, it is often considered to be a "black box" operation whose results are difficult to interpret and sometimes counter-intuitive to the user. In order to assist the user in better understanding and utilizing PCA, we have developed a system that visualizes the results of principal component analysis using multiple coordinated views and a rich set of user interactions. Our design philosophy is to support analysis of multivariate datasets through extensive interaction with the PCA output. To demonstrate the usefulness of our system, we performed a comparative user study with a known commercial system, SAS/INSIGHT's Interactive Data Exploration. Participants in our study solved a number of high-level analysis tasks with each interface and rated the systems on ease of learning and usefulness. Based on the participants' accuracy, speed, and qualitative feedback, we observe that our system helps users to better understand relationships between the data and the calculated eigenspace, which allows the participants to more accurately analyze the data. User feedback suggests that the interactivity and transparency of our system are the key strengths of our approach.

Categories and Subject Descriptors (according to ACM CCS): User Interfaces [H.5.2]: Interaction styles (e.g., commands, menus, forms, direct manipulation)—Methodology and Techniques [I.3.6]: Interaction techniques—

1. Introduction

Principle Component Analysis (PCA) is a widely used mathematical technique for high dimension data analysis. Just within the fields of computer graphics and visualization alone, PCA has been used in many different research areas [Jol02]. At its core, PCA is a method that projects a dataset to a new coordinate system by determining the eigenvectors and eigenvalues of a matrix (Figure 1). This method finds the factors which explain the most variation among data points.

Although PCA is a powerful technique capable of reducing dimensions and revealing relationships among data items, it has traditionally been viewed as a "black box" approach that is difficult to grasp for many of its users [Jol02, Shl05]. The process and result of the coordinate transform from original data space into eigenspace in PCA makes it challenging for the end user to identify the relationships between the input data and the data after the projection into eigenspace. This is especially problematic for novice users and students who need to use PCA but do not yet grasp how it works. Without a certain amount of background knowledge in the math behind PCA, it is often difficult for the user to perform effective analysis both in understanding how the original data items transform between coordinate systems and how the data dimensions relate to the principle components.

In order to assist the user in better understanding and utilizing PCA for analysis, we have developed a system called iPCA (interactive PCA) that visualizes the results of principle component analysis using multiple coordinated views and a rich set of user interactions. The four coordinated views in our system visualize the data items in original data space (Data View), the data items in eigenspace (Eigenvector View), the data items projected onto two principle components (Projection View), and the correlations between all data dimensions (Correlation View). User interactions in one view are immediately reflected in the others so that the user

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

can easily identify a data item or a data dimension in the original data space and its counterpart in eigenspace.

To demonstrate the effectiveness of iPCA, we performed a comparative user study with a well-known commercial system called Interactive Data Exploration, which is part of SAS/INSIGHT. The two systems are similar in that both systems use the same mathematical functions for performing PCA calculations, but they differ in their approaches to interface and interaction design. While the visualizations and interactions in our system are fluid, dynamic, and coordinated, in SAS/INSIGHT, a more traditional menu-driven and command-line approach forms the basis of interaction. Using SAS/INSIGHT, the user iteratively inputs parameters into the system before clicking on a button (or typing in a command) to initiate the PCA process and generate the results as static images and charts.

Our user study involved 12 participants performing complex analysis tasks on high dimensional data using both iPCA and SAS/INSIGHT. We quantitatively measured the accuracy and speed of the users' analyses, and asked the participants for qualitative feedback on ease of use, preference, and effectiveness. Based on the quantitative results of the user study, we find that users were faster and more accurate in analysis tasks using our system. Participants' feedback indicates that our system better facilitates the understanding of PCA, is more intuitive to use, and is unanimously preferred over SAS/INSIGHT. Many participants attributed the success of our system to its high interactivity and transparency, which suggests that our system is successful in opening up the "black box" of principle component analysis.

The rest of the paper consists of six sections. First we discuss other research in visualizing PCA and the benefits of interaction. Then, we provide our system's interface design and the available sets of interactions. In section 5, we introduce the evaluation procedures and results, and conclude with discussions, conclusion, and future work.

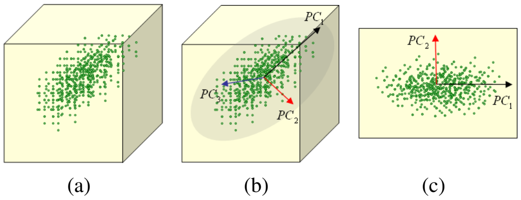


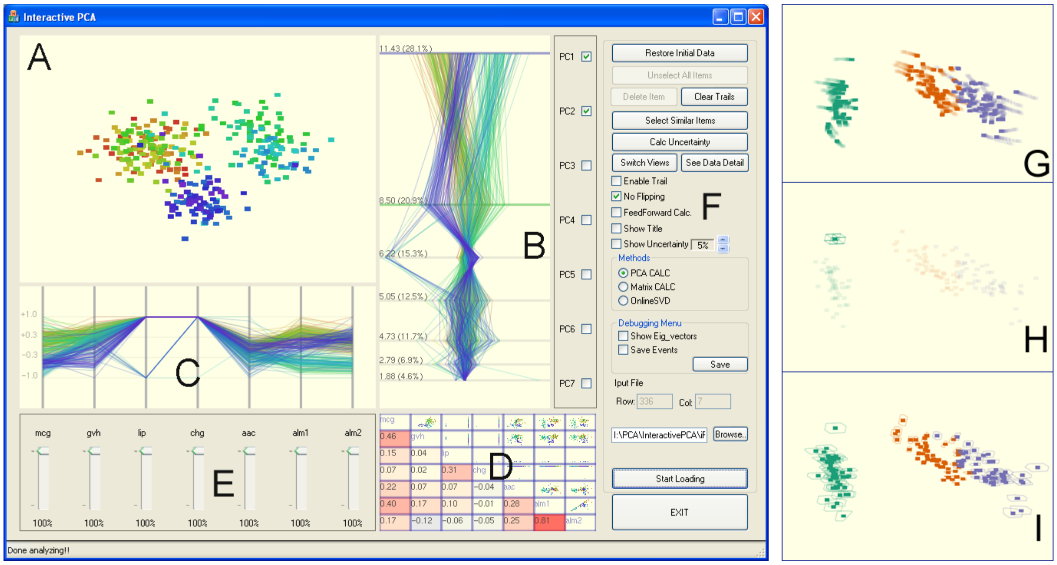
Figure 1: *Illustration of principal component analysis. High-dimensional data (a) are plotted with respect to their first three Principal Components (PCs) (b) and first two PCs (c).*

2. Previous Work

PCA has been applied in many disciplines for various purposes. In visualization, PCA is used mostly for dimension reduction. For example, Hibbs et al. [HDLT05] apply PCA to visually analyze microarray data. Wall et al. [WRR03] demonstrate how to visualize gene expression data using PCA and how to interpret the results. However, while PCA is popular and effective as a tool, there have been few available products or research projects on assisting the understanding of PCA results. Mathematical applications such as MATLAB [Theb] and SAS/INSIGHT [SAS] can perform PCA and visualize its results accurately. An open-source visualization tool, GGobi [Thea], supports interactive analysis of data through PCA and can be linked to R (Statistical Computing Software) for additional statistical methods. Müller and Alexa [MA04] developed a system which allows the user to visually detect and create clusters of data elements in the PCA space. Müller et al. [MNS06] further enhanced conventional information visualizations with PCA and demonstrated that this combination improved data analysis. All these PCA-based tools are powerful and employ various visualization techniques. However, they also share the same goal of utilizing PCA with the assumption that users are experts at mentally transforming data elements from their original space into the projected PCA space. Our work differs in that we intend to use interaction to make the transformation of coordinate spaces intuitive to both novices and experts, and to show that by opening this "black box," users can gain a deeper understanding of data analysis using PCA.

Interaction plays an important role in visualization for assisting users in understanding their data. Several user evaluations have found a benefit for interactive visual systems over traditional iterative input systems in understanding and using data. Ahlberg et al. [AWS92] study the difference between using dynamic sliders and traditional text entry to visually explore periodic table data. They find that participants are faster with the dynamic slider interface on some but not all of their tasks. However, they do not find a clear difference in the participants' subjective evaluation of the interfaces.

More recently, Callahan and Koenemann [CK00] compare an interactive visual tool, InfoZoom, against two traditional interfaces for online catalog browsing. With InfoZoom, users are more likely to complete tasks faster. The users also report higher ease of use and efficiency than traditional interfaces. In contrast, Combs and Bederson [CB] compare a zoomable image browser to a static image browser and find no difference in performance, although users tend to (non-significantly) prefer the zoomable browser. Unfortunately, while these studies inform us of the value of interaction, the tasks are simpler than asking users to perform complex analysis using PCA. Although some research studies [SNLD06, SS06] have been performed to understand the effects of interactions in a more complex analysis task, these studies are narrowed to finding the effectiveness and the limitations of their applications.



☑ PC1
☑ PC2
□ PC3
□ PC4
□ PC5
□ PC6
□ PC7

□ Enable Trail
☑ No Flipping
□ FeedForward Calc.
□ Show Title
□ Show Uncertainty 5%

Methods
● PCA CALC
○ Matrix CALC
○ OnlineSVD

Debugging Menu
□ Show Eig_vectors
□ Save Events

Iput File
Row: 336 Col: 7
I:\PCA\InteractivePCA\iF

mcg						
0.46	gvh					
0.15	0.04	lip				
0.07	0.02	0.31	chg			
0.22	0.07	0.07	−0.04	aac		
0.40	0.17	0.10	−0.01	0.28	alm1	
0.17	−0.12	−0.06	−0.05	0.25	0.81	alm2

Figure 2: *The system overview (left) showing the four views and the two control panels with the E. Coli dataset, and three examples with the Iris dataset (right). (A) Projection view. Data items are projected onto the two user-selected eigenvectors (in this case, the primary and secondary principle components). (B) Eigenvector view. Each eigenvector is treated as a dimension in this parallel coordinates view, and every data item is drawn as a line. (C) Data view. Another parallel coordinates view, but this time each dimension represents the dimensions in the original data, and each line represents each data item. (D) Correlation view. Pearson-correlation coefficient and the relationships (scatter plot) between each pair of variables are represented. (E) Dimension sliders. Each slider controls the amount of contribution of a dimension in the PCA calculation. (F) Control options. (G) shows the result of diminishing the first dimension (Sepal length) of the Iris dataset from 100% to 0%. The trails show how the data points move in PCA space in response to the change. The images (H) and (I) show 10% uncertainty in the data (in all dimensions). The possible locations for each data point are drawn in a hypercube (H) and in outlines (I) corresponding to the number of data item(s) selected.*

3. Interface Design

The overall interface design of our system, iPCA, is based on multiple coordinated views. Each of the four views in the system represents a specific aspect of the input data either in data space or eigenspace, and are coordinated in such a way that any interaction with one view is immediately reflected in all the other views (brushing & linking). The coordination between the views depicts the same data item or data dimension in both data space and eigenspace simultaneously, thus allowing the user to infer the relationships between the two coordinate spaces.

Along with two control panels, iPCA contains four distinct views: the Projection View (Figure 2A), the Eigenvector View (Figure 2B), the Data View (Figure 2C), and the Correlation View (Figure 2D).

Projection View: Two principal components (by default, the first and second most dominant eigenvectors) are used to project data points onto a two-dimensional coordinate system.

Data view: The Data View is located below the Projection View, and shows a parallel coordinates visualization of all data points in the original data dimensions. In this view, an auto-scaling function is applied to increase the readibility of data.

Eigenvector View: In the Eigenvector View, data points are shown in the eigenspace. The calculated eigenvectors and their eigenvalues are displayed in a vertically projected parallel coordinates visualization, with eigenvectors ranked from top to bottom by dominance. The distances between eigenvectors in the parallel coordinate view vary based on their eigenvalues, separating the eigenvectors based on their mathematical weights.

Correlation View: Pearson-correlation coefficients and relationships between variables are represented in the Correlation View as a matrix of scatter plots and values. Since correlations between dimensions are symmetric, repetition is avoided by separating the matrix into three components: the diagonal, the bottom triangle, and the top triangle. The diagonal displays the name of the dimension as a text string. The bottom triangle shows the coefficient value between two dimensions with a color indicating positive (red), neutral (white), and negative (blue) correlations. The top triangle contains cells of scatter plots in which all data items are projected onto the two intersecting dimensions. The colors of the data items are the same as the colors used in the other three views so that clusters are easily identified.

It is relevant to note that the selection operation in all views and the zooming-in mechanism in the Projection and Correlation views help users to focus their interest on a data item or items. Also, the Projection View and the Correlation View can be switched such that the Projection View takes up the lower right hand position and the Correlation View fills the main display. This simple switch operation allows the user to utilize the visual real estate for focusing either on a single projection of data or to examine in detail all (or one) scatter plot(s) in the Correlation View.

The two control panels include a set of dimension sliders (Figure 2E) that can be used to decrease or increase the contributions of each of the original data dimensions, whose purpose will be discussed further in the following section (Section 4). Several additional modes can also be specified in the other control panel to enhance understanding of the visual changes during data analysis (Figure 2F). The user can enable *trails* so that the path of each data point's recent motion is painted to the screen, making the movement of each point during interaction operations more apparent. The user can also choose to show *uncertainty* (Figure 2H and I) by setting a percentage of possible error in the dataset, which is reflected as bounding boxes around data items in the Projection View.

4. Interaction

Since iPCA is designed with high interactivity in mind, the types of available interactions are carefully considered. We categorize all the interactions in iPCA into two groups: interactions with the views, and interactions with PCA. Interactions with the views are operations that do not result in PCA calculations, and include brushing, filtering, zooming and panning, etc; whereas interactions with PCA will result in new PCA calculations, including operations that change the weights of dimensions, move data points in either data space and eigenspace, and removal of data points. Both types of interactions are embedded in the coordinated views such that all views react to all interactions.

4.1. Interacting with the Views

Interactions in this category are operations that do not cause the system to recompute PCA. As mentioned above, these operations include brushing, filtering of data items or dimensions, zooming and panning, etc. Although these interactions are standard in most Infovis or visual analytics tools, they are nonetheless very important, and are essential in multiple coordinated views. The ability to allow the user to select a cluster of data items in one coordinate space and immediately see the corresponding items highlighted in the other coordinate space helps the user understand the relationship between the two.

The most notable interactions in this category are the different types of selections implemented in iPCA. iPCA allows the user to select data items in all four views. In Data View and Eigenvector View, where the visualizations are parallel coordinates, selection means clicking on a single line or brushing a range of items. In Projection View and Correlation View, the user can either click on a single dot or draw an enclosed space upon which all data items within the space will be selected.

4.2. Interacting with PCA

As mentioned previously, one of the biggest hurdles in effectively analyzing PCA results is in understanding the relationships between data space and eigenspace. While the interactions provided in the previous section allow the user to see a data item appear in different coordinate systems, the interactions do not immediately lead the user to see the relationship between the coordinate spaces. Specifically, eigenvectors are linear combinations of data dimensions, therefore, understanding which data dimension contributes to an eigenvector is a key point in comprehending how the coordinate spaces relate to each other.

In order to visually assist the user in recognizing how data space relates to eigenspace, we create a set of interactions that allow the user to alter the values of the data items. For example, if the user drags a data item in the Projection View towards the positive direction along the x-axis (increasing the data points value in the first principle component), the user should be able to immediately observe in the Data View how that change affects the values of that data item in the original data space, thus shedding light on the relationship between the first principle component and all dimensions in the original data space.

Similarly, if there is an obvious cluster in the Projection View, the user can interactively change the weights of a dimension to see its affect on the formation of the cluster. For example, if diminishing the contribution of a data dimension in PCA calculation down to 0% does not affect the clustering, then it should be clear that the cluster does not depend on that particular dimension.

While the concept of encouraging interactions that directly alter the values of data items seem counter-intuitive, the idea is not novel. *Spotfire* includes a "jitter" operation [Ahl96], and *Dust and Magnet* has a "dust shake" operation [YMSJ05], both of which are designed to reveal occluded data items. In medical visualization, deformation or "cut-aways" modify the data to expose hidden structures underneath skin and flesh [MTB03]. The interactions in iPCA share a similar goal, but instead of revealing hidden or occluded information, our interactions assist the user in revealing relationships between coordinate spaces.

Three specific interactions are implemented based on the concept of data alteration: modifying dimension contribution, adjusting data items, and removal of data items.

Modifying Dimension Contribution: Each slider in Figure 2E corresponds to a data dimension. By modifying the slider, the user can change the contribution of the data dimensions in the final PCA calculation. For instance, changing the dimension contribution to 50% indicates the weight change of the selected dimension to 0.5. This interaction allows the user to observe which data dimensions contribute to the projections of the data in eigenspace. By adjusting these sliders, a user can quickly test hypotheses about how the analysis would be affected if a dimension or set of dimensions were removed or considered less important. This makes it possible for a user to observe the formation and dispersion of clusters and to identify the cause of outliers.

Adjust Data Items: Values of data items can be modified in either the Projection View, Data View, or Eigenvector View. This interaction not only allows the user to see the relationship between a principle component and the contributing data dimensions as mentioned above, but also allows the user to test what-if scenarios. If the user suspects that a data item should appear in a certain cluster, the user can manually move the data item and see how the values of that data item would have to be modified.

Removing Data Items: In analysis using PCA, a common task is for the user to remove outliers. iPCA supports direct removal of data items from the system so that the user can observe how the projection from data space to eigenspace changes with the removal.

One caveat of these interactions is that they are computationally expensive. Modifying any data requires the re-computation of PCA, and in the cases of interactively adjusting sliders and moving data items on screen, PCA has to be re-calculated quickly to avoid lag or flickering. For very large datasets, this type of interactions has the potential of becoming a bottleneck in usability. In iPCA, the scalability issue is addressed by incorporating a faster version of singular value decomposition called online-SVD [Bra06] which trades precision for speed. Brand demonstrates how online-SVD is faster than traditional SVD (see [Bra06] for detail). The user has the option to use either traditional SVD or online-SVD depending on the speed and accuracy requirements as well as the scale of the data.

5. Evaluation

We conducted a comparative evaluation to assess the effectiveness of our system in relation to a well-known commercial tool, SAS/INSIGHT's Interactive Data Exploration. A total of 12 students (nine males) participated in the evaluation. Three of the 12 participants were undergraduate students and nine were graduate students, and 11 of the participants majored in computer science and one in management of information science. Based on self reported familiarity, we found that nine participants were aware of PCA prior to the evaluation, and of the nine, three had used PCA in the past.

At the start of the evaluation, all participants receive a detailed explanation about PCA followed by a pre-evaluation background questionnaire. Each participant was provided a total of ten minutes to train with the two systems prior to the evaluation. The evaluation consisted of performing four analysis tasks using each system. The participants were given five minutes to perform each task and were requested to answer questions immediately after each task. The evaluation was conducted using an online website, where time spent and answers were saved into a database.

We performed the evaluation using three different datasets: the Iris dataset (150 data items × 4 dimensions), the E.coli dataset (336 data items × 7 dimensions) and the Wine dataset (179 data items × 13 dimensions). The Iris dataset was used in the training session whereas the E.coli and the Wine datasets were used in the actual evaluation. All three datasets are scientific results that are publicly available at the UCI Machine Learning Repository [AN07].

5.1. Procedure

Each participant was requested to use the two systems on different datasets. Therefore, six participants used iPCA first and the rest of the participants began with SAS/INSIGHT. The order in which datasets were given to each participant was counterbalanced with system order, so that six participants used the E.Coli dataset first and the rest used the Wine dataset first.

Four tasks were given to each participant during the evaluation of both systems:

- What is the most striking outlier you can find? An outlier is a point that does not fit the overall patterns of the dataset.
- Find a dimension that least affects the PCA outputs in the Projection View using first and second principle components.
- Find two dimensions with a highly positive correlation. Also find the class name and label of an outlier that does not follow that correlation.
- How does removing the first dimension affect the PCA results using the first and second principle components? List as many observations as possible.

The first three tasks are related to finding exact answers and the last one is a descriptive task asking the participant to describe the difference between including and excluding a specific dimension. Five minutes were given to solve each task. If time expired, partial answers were saved into the database. As soon as each task was completed, a post-task questionnaire was given to participants to track how they

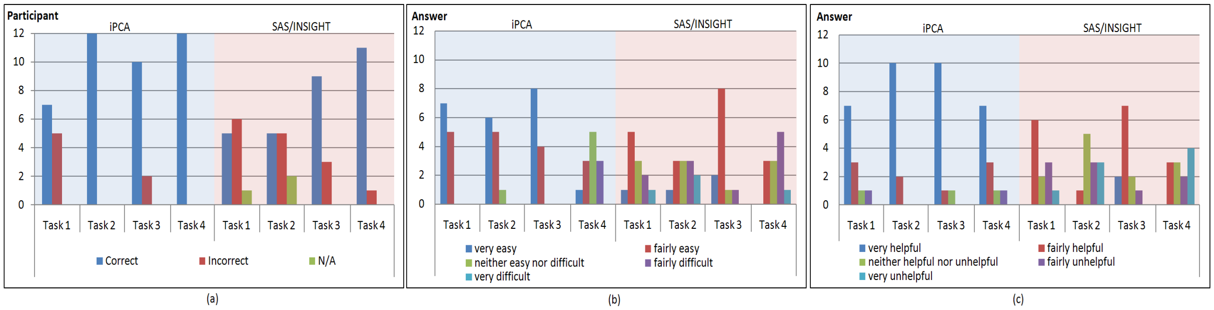


Figure 3: *Results broken down by tasks for each of the two systems. (a) Number of participants who answered the task question correctly. (b) Task difficulty and (c) helpfulness of the system in solving the task, as reported by participants.*

felt about the task. These questions included "How difficult was this task?" and "How helpful was the interface in solving the task?" A post-application questionnaire was given after a participant completed all four tasks. This questionnaire asked the participant to give feedback on their overall subjective opinion about each system. After a participant completed the evaluation using both systems, the participant completed an additional set of questions (post-study questionnaire) that described the preference, the ease of use, and the effectiveness of the system in analyzing data. Finally, the participant graded each system on a scale of 'A' to 'F'.

5.2. Results

We present the results of our evaluation based on accuracy, speed, difficulty and usefulness, effectiveness, and preference. Both accuracy and speed are measured quantitatively; whereas the other three categories are analyzed based on the participants' qualitative feedback.

Accuracy: Figure 3(a) shows the results of the participants' accuracy in solving each task using both iPCA and SAS/INSIGHT. As shown, approximately 85% of the participants answered correctly using iPCA. On the other hand, when using SAS/INSIGHT, they were only able to answer correctly 62% of the time. Furthermore, when using SAS/INSIGHT, there were three instances in which a participant could not complete the task. One of the instances was due to the fact that the participant ran out of time. In the other two cases, the participants simply gave up and claimed that they were unable to find the solutions (see Figure 3(a)). Note that the accuracy difference is statistically significant across the two systems ($p < 0.01$).

Speed: Table 1 shows the overall average time spent solving each task. Participants spent less time in solving each task using iPCA except for task 4 (a descriptive question). This seems to be because participants tried to find as many differences as possible through interaction with dimensions.

On average, participants spent about 150 seconds using iPCA, and 170 seconds using SAS/INSIGHT. Although the difference is not statistically significant ($p = 0.17$), there is a trending effect towards a faster solution when using iPCA.

Difficulty & usefulness (post-task questionnaire): Figure 3(b) and (c) show how participants rated the difficulty of each task when using iPCA or SAS/INSIGHT, as well as how they rated the usefulness of each system in solving the task. Figure 3(b) indicates that about 81% of the participants found the tasks to be easy when using iPCA. On the other hand, only about 48% of the participants identified the tasks as being easy when using SAS/INSIGHT. Interestingly, although more than half of the participants mentioned that task 1 is easy to solve, Figure 3(a) indicates that the accuracy in solving task 1 is low (58% iPCA and 41% SAS/INSIGHT). This might be because most participants have little previous experience with finding outliers.

Figure 3(c) shows that about 90% of the participants identified iPCA to be helpful in solving the tasks; whereas only about 40% of the participants found SAS/INSIGHT to be helpful. Furthermore, only two participants (participant D and I) rated iPCA to be not helpful in solving a task (tasks 4 and 1, respectively); whereas ten participants indicated that SAS/INSIGHT was unhelpful in solving some tasks (one participant indicated SAS/INSIGHT was completely not helpful in solving all tasks).

Overall, we find that the more difficult a task was rated (very easy = 5, fairly easy = 4, etc), the more time the participants spent on solving it ($p < 0.0001$). However, solving a task with a "helpful" system (very helpful = 5, fairly helpful = 4, etc) did not decrease the time spent on the task

Table 1: *Average time spent in solving each task.*

Application	Task	Time Spent (seconds)
iPCA	Task 1	136.58
	Task 2	128.33
	Task 3	125.50
	Task 4	211.33
SAS/INSIGHT	Task 1	177.67
	Task 2	165.58
	Task 3	142.92
	Task 4	197.08

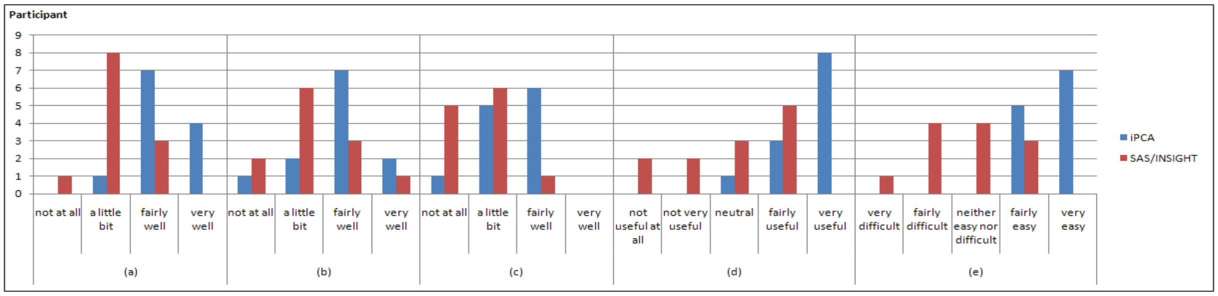


Figure 4: *Participants' responses to a post-application questionnaire, filled out after solving all four tasks using one of the systems. (a) How well do you understand the application now? (b) How well do you understand PCA now? (c) How well do you understand the data you worked with now? (d) How useful was the system? (e) How difficult or easy was the system to learn?*

($p = 0.2586$). With a helpful system, the participants did solve the tasks more accurately ($p = 0.0027$), but participants did not rate the tasks to be less difficult ($p = 0.0966$).

Effectiveness (post-application questionnaire): Figure 4 shows the results of the five questions in the post-application questionnaire conducted right after the evaluation of each system. Of particular significance are the questions asking the participants how well they understood the application (Figure 4(a)), how well they understood the data (Figure 4(c)), how useful was the system (Figure 4(d)), and how difficult or easy was the system to learn (Figure 4(e)).

In answering how well the participants understood PCA, most participants did not indicate that they understood PCA "very well." However, the majority of the iPCA users indicated that they understood PCA "fairly well"; whereas the majority of the SAS/INSIGHT users only claimed "a little bit" of understanding.

In answering how well the participants understood the data, the majority of the iPCA users indicated that they understood the data either "fairly well" or "a little bit"; whereas the SAS/INSIGHT users either understood the data "a little bit" or "not at all."

Lastly, in answering about the usefulness of the system, the majority of the iPCA users found the system to be "very useful"; whereas SAS/INSIGHT users consistently ranked the system to be "fairly useful" and below, with four participants claiming the system to be "not very useful" or "not useful at all."

Preference (post-study questionnaire): After the evaluation, each participant ranked the two systems and described their pros and cons. Figure 5 clearly shows that most participants preferred iPCA over SAS/INSIGHT, giving iPCA eight A's and four B's. On the other hand, the majority of the participants gave SAS/INSIGHT a C or D grade, with one participant failing it by giving it an F.

When describing the pros and cons of iPCA, eight participants specifically pointed out the strength of iPCA as being "interactive," and eight participants described iPCA as "transparent." While a few participants gave constructive feedback on how to further enhance the iPCA tool (e.g., add the ability to rearrange the dimensions in the Correlation View), the only negative criticism for iPCA was that it did not generate printable reports similar to the static charts and numbers that SAS/INSIGHT generates.

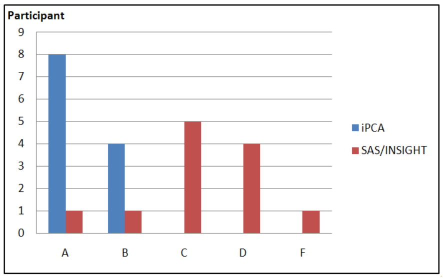


Figure 5: *At the end of the evaluation, each participant grades the systems on a scale of 'A' to 'F'.*

For SAS/INSIGHT, two participants who were previously familiar with the SAS system pointed out that while they preferred iPCA over SAS/INSIGHT for analyzing PCA results, SAS is still a far more comprehensive and complete numerical and statistical analysis tool. One participant further noted that if he was allowed to use additional features in SAS outside of the tools specific to PCA, deeper analysis on the dataset could have been performed.

6. Discussion

Since our evaluation compared two systems (iPCA and SAS/INSIGHT) that use the same mathematical methods for computing PCA, we can safely assume that the increase in our participants' performance in using iPCA is attributed solely to the interface design and the set of interactions. Unfortunately, we are not able to further isolate the specific factor(s). Based on our evaluation alone, we cannot determine if the increase is due to the multiple coordinated views, the interactions, or the combination of the two. However, we do hypothesize that the "interactions with PCA" play a significant role in that the user's direct and continuous manipulation with PCA is rewarded with immediate visual feedback. This allows the user to "play" with the data and intuit the subtleties behind the coordinate transform between data space and eigenspace in a way that less interactive visualizations such as SAS/INSIGHT cannot achieve.

The "interactions with PCA" are also the most unique set of the interactions in iPCA. Unlike our other interactions that merely highlight or explore the data, the design decision behind the "interactions with PCA" is to focus on reasoning. In fact, we design the "interactions with PCA" to be less faithful to the data, but more revealing in discovering relationships between coordinate spaces and data dimensions. For example, most of our participants credited the rich interactions in iPCA to be the primary strength of the system, but two of our participants pointed out the fact that in modifying dimension contribution, moving a slider from 100% to 72% and taking a snapshot of the Projection View was not meaningful as the projection was not of the original data. Similarly, moving a data point across the screen seemed counter-intuitive as it directly modified the values of the data. While these concerns are valid, we contend that they miss the spirit of the interactions. It is true that the resulting images from these interactions cannot be considered by themselves, but it is during the direct manipulation of the data and coordinate spaces that the user gains insight about their relations and how changes in one affects the other, which is otherwise hidden. One very interesting future direction for our research will be to further understand why these types of interactions are successful, and examine the extent to which they can be applied.

7. Conclusion and Future Work

We present a visual analytical system for analyzing PCA results called iPCA. We design the interface using multiple coordinated views, and add a rich set of interactions for both interacting with the views and interacting with the PCA calculations. To validate the effectiveness of our system, we performed a comparative user study with a well-known commercial system called SAS/INSIGHT. The participants of the evaluation used both iPCA and SAS/INSIGHT to perform complex analysis with high dimensional datasets. The results of the evaluation indicate that iPCA is somewhat faster, more accurate, easier to use, more effective in learning about PCA and the dataset, and is overwhelmingly preferred over SAS/INSIGHT.

Since iPCA and SAS/INSIGHT use the same mathematical functions in performing PCA, the difference in the evaluation between the two systems can only be attributed to either the interface design or the interactions. While our current evaluation cannot isolate the specific factor(s), we can gather some important insights and have a significant basis for further studies.

References

[Ahl96] AHLBERG C.: Spotfire: An information exploration environment. *SIGMOD Record 25*, 4 (1996), 25–29.

[AN07] ASUNCION A., NEWMAN D.: UCI machine learning repository, 2007.

[AWS92] AHLBERG C., WILLIAMSON C., SHNEIDERMAN B.: Dynamic queries for information exploration: an implementation and evaluation. In *SIGCHI* (1992), ACM.

[Bra06] BRAND M.: Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications 415*, 1 (2006), 20–30.

[CB] COMBS T. T. A., BEDERSON B. B.: Does zooming improve image browsing? In *ACM Conference on Digital Libraries*, pp. 130–137.

[CK00] CALLAHAN E., KOENEMANN J.: A comparative usability evaluation of user interfaces for online product catalog. In *EC '00* (2000), ACM Press, pp. 197–206.

[HDLT05] HIBBS M. A., DIRKSEN1 N. C., LI K., TROYANSKAYA O. G.: Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics 6*, 115 (2005).

[Jol02] JOLLIFFE I. T.: *Principal Component Analysis*, second ed. Springer, 2002.

[MA04] MÜLLER W., ALEXA M.: Visual component analysis. In *VisSym 2004* (2004), Eurographics Association, pp. 129–136.

[MNS06] MÜLLER W., NOCKE T., SCHUMANN H.: Enhancing the visualization process with principal component analysis to support the exploration of trends. In *APVis '06* (2006), Australian Computer Society, Inc., pp. 121–130.

[MTB03] MCGUFFIN M. J., TANCAU L., BALAKRISHNAN R.: Using deformations for browsing volumetric data. In *IEEE Visualization* (2003), IEEE Computer Society, pp. 401–408.

[SAS] SAS INSTITUTE, INC: SAS/INSIGHT. http://sas.com/technologies/analytics/statistics/insight.

[Shl05] SHLENS J.: A tutorial on principal component analysis. http://www.snl.salk.edu/~shlens/notes.html, 2005.

[SNLD06] SARAIYA P., NORTH C., LAM V., DUCA K. A.: An insight-based longitudinal study of visual analytics. *IEEE TVCG 12*, 6 (2006), 1511–1522.

[SS06] SEO J., SHNEIDERMAN B.: Knowledge discovery in high-dimensional data: Case studies and a user survey for the rank-by-feature framework. *IEEE TVCG 12*, 3 (2006), 311–322.

[Thea] THE GGOBI FOUNDATION, INC: Ggobi. http://www.ggobi.org.

[Theb] THE MATHWORKS, INC: Matlab. http://www.mathworks.com/products/matlab.

[WRR03] WALL M. E., RECHTSTEINER A., ROCHA L. M.: Singular value decomposition and principal component analysis. In A Practical Approach to Microarray Data Analysis (D.P. Berrar, W. Dubitzky, M. Granzow, eds.) Kluwer: Norwell, MA, pp. 91-109., 2003.

[YMSJ05] YI J. S., MELTON R., STASKO J., JACKO J. A.: Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization 4*, 4 (2005), 239–256.