Learning Cluster Analysis through Experience

Joshua M. Lewis
josh@cogsci.ucsd.edu
Department of Cognitive Science
University of California, San Diego

Virginia R. de Sa
desa@cogsci.ucsd.edu
Department of Cognitive Science
University of California, San Diego

Abstract

The field of machine learning is constantly developing useful new techniques for data analysis, but they are often ignored by researchers outside the field due to unfamiliarity and the difficulty of keeping up with a large body of work. We propose a methodology for training researchers how algorithms work through experience, such that they gain an implicit, rather than explicit, understanding of their function. Thus we combine theory from discovery learning with advanced software and a more educated target population to foster such understanding. We have developed an open source application for exploratory data analysis called Divvy that lets users quickly and visually interact with a range of data analysis techniques. Using a simplified version of Divvy, we find that undergraduate subjects are generally able to learn machine learning concepts through experience, though they have only partial success in applying them.

Keywords: unsupervised machine learning; clustering; discovery learning; human computer interfaces

Machine learning has a PR problem. The field has developed many techniques that cluster, classify, or reduce the dimensionality of data, and most techniques could be profitably applied to scientific data sets. Researchers that are not machine learning experts face a daunting question, however–which techniques should I use to analyze my data? Authors proposing a new technique will focus on its strengths over its weaknesses, and most researchers do not want to spend a year reading math papers and becoming a machine learning expert in order to best analyze their data. So too often the analysis technique used is the convenient one (freely available online or as part of a software package), or the traditional one. Researchers miss out on the advances in machine learning, and the machine learning field is not as valuable as it could be to the broader scientific community.

There are two fundamental problems: expertise and access. Gaining expertise is difficult–if a researcher wants to find the right technique for the job, but is unwilling to engage in the time-consuming process of learning the details of every technique, how can they be trained to apply the best one? With the right tools, we believe discovery learning has substantial potential for training researchers. Software that provides direct and intuitive access to the behavior of machine learning algorithms can support the development of a pragmatic (not mathematical) understanding of the algorithms.

As an analogy, baseball players have an excellent idea of how baseballs behave. A baseball's behavior is, of course, governed by the laws of physics and an explicit description of that behavior might be quite complex when spin, deformation, wind and field texture are taken into account. Nevertheless, through extensive experience baseball players acquire an excellent pragmatic understanding of how baseballs behave, an understanding that one might guess is founded on an implicit learned model of baseball behavior rather than the explicit model a physicist would give. We believe that interactive experience with machine learning techniques can give rise to a similar sort of practical and implicit model of algorithm behavior, and that researchers can use such a model to make informed decisions during data analysis.

Discovery learning is particularly compelling in this context because researchers often do not have the time or inclination to seek out traditional forms of instruction while analyzing data. Tools that support learning on the job are thus necessary and expedient. In this paper we test our hypothesis using a data analysis platform called Divvy that we've developed to provide such an experience that emphasizes speed and visualization.

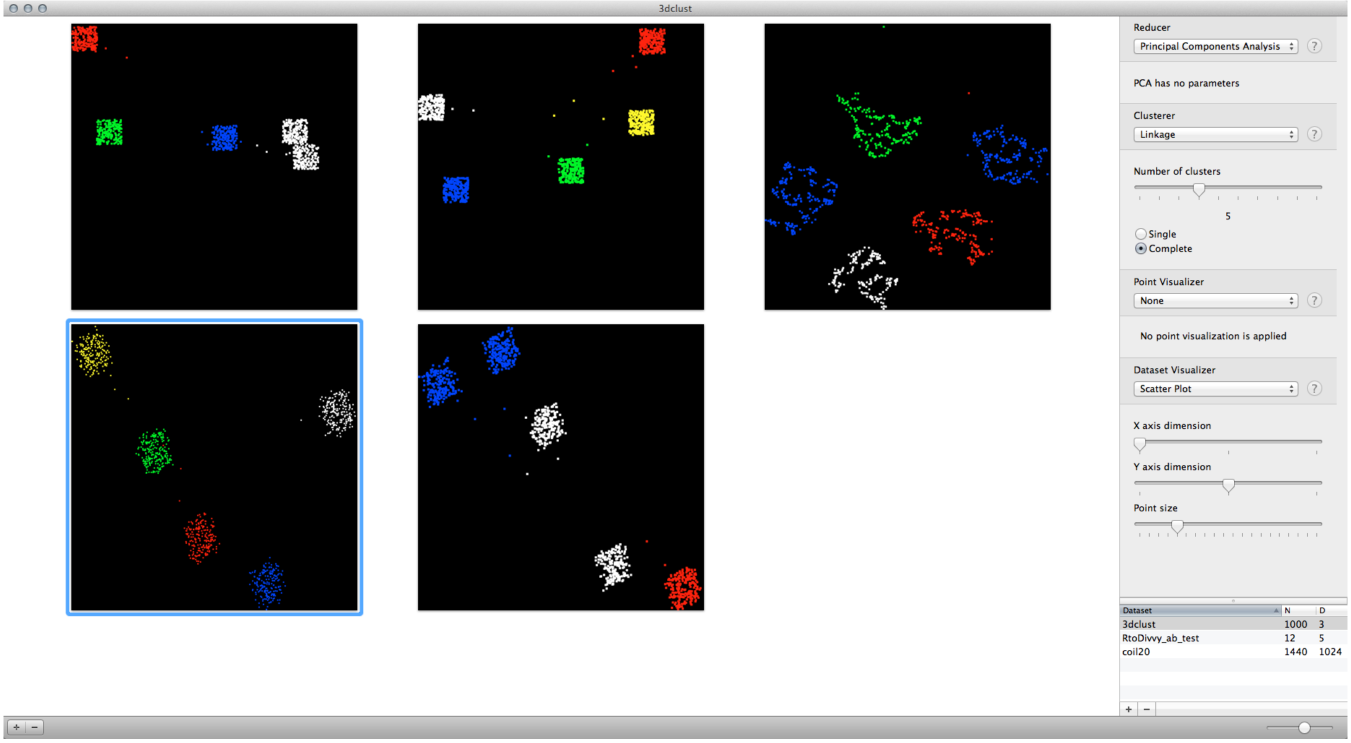
Gaining access to a wide variety of data analysis techniques is also tricky–it might require technical knowledge (e.g. basic programming skills in whichever languages the techniques are in), or owning proprietary software like Matlab and formatting one's data for it. To that end Divvy is a free, open source project designed around a plugin architecture where machine learning researchers can package their algorithms with intuitive custom UIs that require no programming expertise from users.

In our experiment we give undergraduate subjects interactive experience with two clustering techniques, *k*-means (MacQueen, 1967) and single linkage (Johnson, 1967), labeled simply as method A and method B and without any explicit instruction as to their differences. We find that after training almost every subject learns a few relevant facts about A or B or their parameters, and that some subjects appear to be able to apply this knowledge to new analysis contexts.

Divvy

Data analysis is often a laborious process. A researcher collects data, and then loads it into a software package such as Matlab or R. To apply an algorithm to his or her data, the researcher has to write a command or fill out a dialog box and then wait for processing to finish. Finally, the researcher will use other commands to visualize the algorithm's output. To change a parameter and see the impact it has, this process must be repeated. Some researchers might write a script that runs a set of different parameters and visualizations, and then go out for a coffee and come back to see if the whole endeavor bore any fruit.

This is a tenuous kind of interaction. A baseball, by virtue of being in the real world, provides critical instantaneous



Reducer
Principal Components Analysis

PCA has no parameters

Clusterer
Linkage

Number of clusters
5

○ Single
● Complete

Point Visualizer
None

No point visualization is applied

Dataset Visualizer
Scatter Plot

X axis dimension

Y axis dimension

Point size

Dataset	N	D
3dclust	1000	3
RtoDivvy_ab_test	12	5
coil20	1440	1024

Figure 1: The full Divvy UI. Each visualization represents a different view of the same dataset (generated by combining a dimensionality reduction technique, a clustering technique and a dataset visualizer) and users can set the properties of each view using the tools to the right. A list of datasets resides in the bottom right, allowing the user to switch between them at any time, even while results are computing in the background.

feedback to those interacting with it. In the above process the algorithm does not, and the goal of the Divvy project is to close that gap and provide an interface where visualization happens instantaneously and researchers can tweak parameters and see their effect in real time. In a way, Divvy is providing the human analog to active learning (Cohn, Ghahramani, & Jordan, 1996), where learning algorithms choose which training samples to get based on what they predict to be the most informative. Divvy is similar in spirit to a tool called GGobi (*Ggobi data visualization system*, n.d.), which brings cutting edge methods in high-dimensional data visualization to a user friendly graphical interface but without the strong machine learning component Divvy provides.

Divvy supports four types of plugin: clusterers, reducers, point visualizers and dataset visualizers. Clusterers and reducers represent clustering and dimensionality reduction algorithms, respectively. Point visualizers represent single points in the dataset with visualizations, such as the image of a handwritten digit, and dataset visualizers represent the entire dataset, for example with a scatter plot. Each view of the dataset (of which a user can have a practically unlimited number) represents a combination of these four plugin types, so a user can compare, e.g., *k*-means in the first two PCA dimensions with spectral clustering in the same embedding.

Divvy achieves real time responsiveness on many datasets through parallel computing. Many personal computers (and all Macs) ship with multi-core processors (CPUs), as well as graphics processors (GPUs) that can be used for general purpose computation. High performance computing research has so far focused on how these hardware resources can make very large problems tractable (Raina, Madhavan, & Ng, 2009). With Divvy, we are using these technologies to make medium problems very fast–fast enough to feel real time, and to invite the exploratory interaction that we believe leads to learning. Even if an algorithm takes a while to run, users can continue to use Divvy to perform other analyses on the same dataset or even on others while they wait. Our UI design puts a focus on visualization, allowing users to simultaneously visualize many perspectives on their data. Algorithm parameters are controlled with standard UI elements (such as sliders or check boxes) rather than having to be specified with code. See Figure 1 for the full Divvy UI and Figure 2 for the simplified version of the UI we used in this experiment.

Divvy does not attempt to replace a user's data analysis workflow, but rather to be a part of it. It can export data and visualizations in standard formats and import from other popular tools. Divvy, its source code, sample datasets, and R/Matlab data importers are freely available from `http://divvy.ucsd.edu` and on the Mac App Store.

Discovery Learning

Our study represents a form of discovery learning (Bruner, 1961), also known as constructivist, inquiry or experiential learning. In discovery learning students learn material independently of explicit instruction by exploring environments, solving problems, or performing experiments. Several researchers have called into question the effectiveness of pure discovery learning, suggesting that active guidance from an instructor (Mayer, 2004), or a sufficient foundation of domain knowledge (Kirschner, Sweller, & Clark, 2006) are required for constructivist approaches to be successful.

Our target audience for Divvy differs from the traditional subjects used in studies of discovery learning. We intend for Divvy to be used by researchers such as faculty and graduate students who have a highly sophisticated understanding of their problem domain. Further, they are accustomed to self-directed learning. In this sense, though they do not have a detailed understanding of machine learning, they do have a foundation of domain knowledge with which they can determine whether the output of a machine learning algorithm is appropriate or not. In addition, Divvy provides some forms of active guidance. Divvy plugin UIs default to reasonable ranges for parameter settings and every plugin can specify a help link that takes users to a relevant resource on the web, such as a paper describing the method or a relevant Wikipedia article.

For these reasons we believe Divvy to be more likely to succeed than other examples of discovery learning that focus on elementary-, middle-, and high-school populations with less active guidance. In this study we use an undergraduate population that is generally less knowledgeable than our target population, representing a more challenging domain than that which Divvy will have in the wild. If undergraduates are able to learn machine learning concepts with Divvy then graduate students, postdocs, and faculty likely can as well.

As outlined above, we believe that guided learning is not necessarily practical or expedient for our target population. So while explicit instruction would certainly allow subjects to learn machine learning concepts, we do not compare Divvy to that form of learning in this paper. Here we focus on what, if anything, subjects are able to learn from a version of our more pragmatic approach to solving machine learning's PR problem.

Methods

We recruited 22 undergraduate subjects for this experiment. Subjects received course credit for participation. One subject was excluded from the study after he indicated at the end during the interview segment that he must not have understood the instructions, and so we analyzed the data from a grand total of 21 subjects.

Each subject performed 36 trials, which were split into two 18 trial blocks, a training block and a testing block. In both blocks, subjects use the sliders to change the number of clusters, k, and the relative weighting of the horizontal and vertical axes in order to best group the points in each stimulus (one stimulus per trial) and then indicate their satisfaction with the result (ranging from 1, not satisfied, to 7, very satisfied). In the training block, subjects use both A and B (k-means and single linkage, respectively) to group the points, and are required to arrive at a solution for each method. In the testing block, neither A nor B are initially selected and subjects must choose which method they want to use for that trial. Once the choice is made they cannot switch. We divided subjects into two groups of 10 and 11. One group's training set was the other's testing set, and vice versa. At the end of the two blocks, subjects filled out an interview form that assessed their knowledge. The eight interview questions were as follows (where circles means the individual data points):

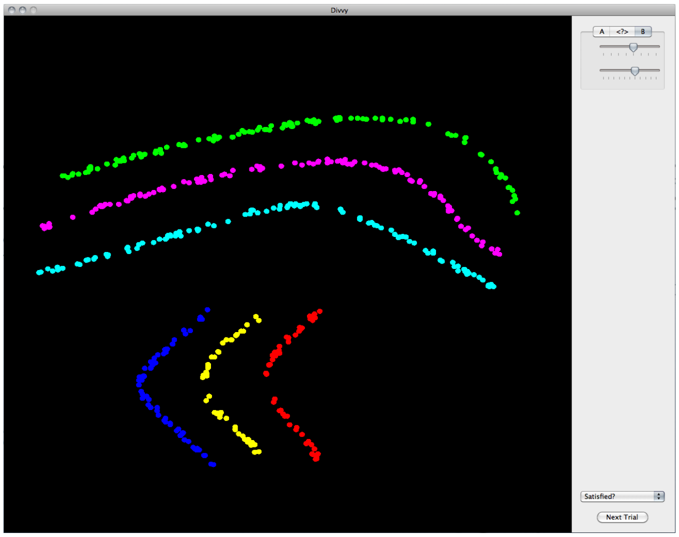


Figure 2: The Divvy UI used in this experiment. The tabs at the top right select method A (k-means) or B (single linkage), and the sliders below control the number of clusters and the relative weighting of the horizontal and vertical axes. Subjects indicate their satisfaction with a particular partitioning using the dropdown menu above the next trial button at the bottom right.

1. What did you feel like method A was doing?
2. What organizations of circles was method A good for grouping?
3. What did you feel like method B was doing?
4. What organizations of circles was method B good for grouping?
5. Did you have a preference between A and B?
6. Why or why not?
7. What did the first (top) slider do?
8. What did the second (bottom) slider do?

We instructed subjects to do their best to learn what A, B and the sliders were doing in the first half of the experiment, as they would need to use that knowledge during the second half. We also made clear that not every stimulus could be ideally grouped with both A and B, and that if they did not like a solution they could just indicate dissatisfaction using the dropdown above the next trial button. We provided two helper images along with the instructions. One showed a well-separated mixture of Gaussians where each Gaussian had its own color. This was held up as a positive example. The second showed two circular groups split in half with color, which was considered a negative example. Beyond these very simple prompts (show in Figure 3) we did not bias the subjects as to what a group should be.

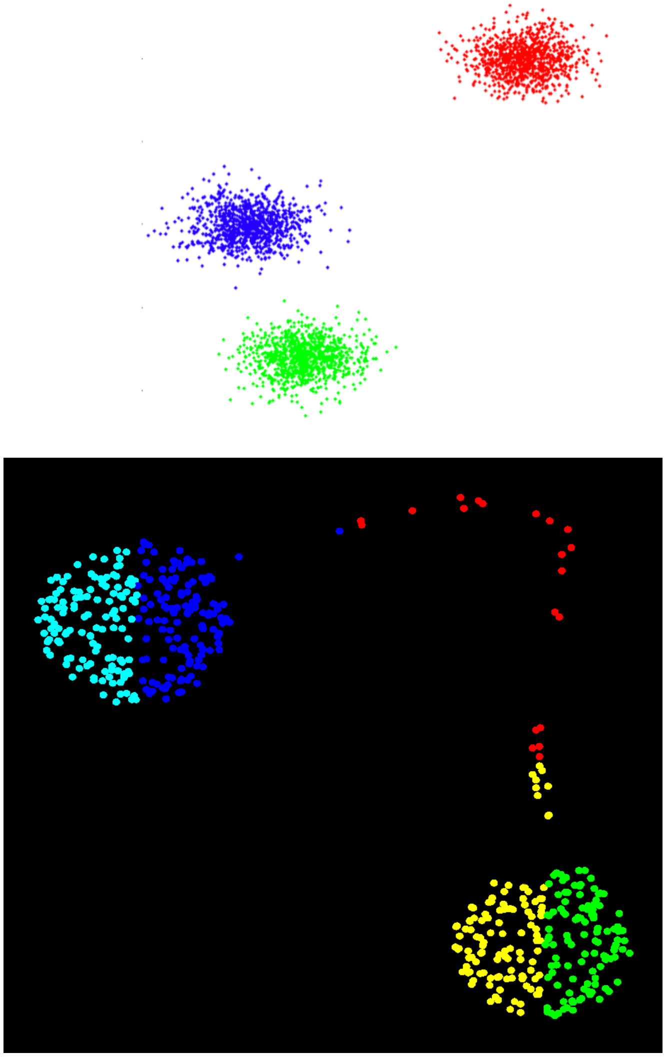


Figure 3: Sample images to give subjects basic guidance on good groups (top) versus bad groups (bottom).

The 36 stimuli fall into three categories, those where A is most effective (14), those where B is most effective (15), and those where A and B are similarly effective (7). We created all 36 stimuli by hand in order to ensure that the first two categories had sufficient membership. Stimuli ranged from complex collections of lines, rings and spirals to connected and disconnected blobs to uniform noise. While these are not real data, so to speak, they provide us with a solid foundation on which to train and judge our subjects that real data would not necessarily provide. Additionally, most meaningful real data are more than two dimensional, and while the full version of Divvy uses dimensionality reduction techniques and multiple views to visualize such data, those techniques are not relevant to our core question in this experiment concerning cluster analysis.

Divvy records every method and parameter combination subjects try over the course of the experiment, including their final grouping and satisfaction. We use these data in concert with interview responses to determine what subjects were able to learn from their experience. From the Divvy records we extract two variables per subject, the total number of different algorithm and parameter settings queried in the training period (the number of "moves"), and the percent of correct method (A or B) choices (with any parameter choice) in the testing period, out of the stimuli for which there is a preferred method. From the interviews we code for understanding of seven possible concepts. The seven possible concepts are as follows:

1. The first slider controls the number of colors (i.e. clusters).

2. The second slider controls the orientation of the boundary between clusters.

3. *k*-means works well on blobs of points (compact regions).

4. Single linkage works well on extended shapes like lines or rings (non-compact regions).

5. *k*-means can work when there is no space separating clusters.

6. Single linkage works best when there is lots of space between clusters.

7. *k*-means tends to divide the points into evenly sized groups, whereas single linkage can make large and small groups.

We hypothesize that there will be a positive correlation between understanding a greater number of concepts and selecting the correct method. We also report correlations between these measures and the number of moves subjects take. To gain an understanding of the relative difficulty of learning the concepts we report in detail the concepts learned on a per-subject basis. Finally, we compare subject satisfaction when using the correct method on a stimulus versus the incorrect method. This test indicates whether subjects recognize when the partitions are not ideal. If the subjects cannot distinguish good partitions from bad given their intuition and the instructive samples, then there is not an opportunity for learning.

Results

In Table 1 we summarize the contents of each subject's interview, using the seven concepts described above. Nineteen of 21 of the subjects learned at least one concept, and 15 of the subjects learned at least one concept excluding the simplest one (the function of the first slider). On average subjects learned 2.4 concepts over the course of the study.

Table 1: A summary of the concepts subjects learned. Subjects in bold chose the correct method for over 70% of stimuli in the test block.

Subjects	1st Slider	2nd Slider	*k*-means Blobs	Single Linkage Shapes	*k*-means No Separation	Single Linkage Separation	*k* Even vs SL Uneven	Sum
1								0
2								0
3	✓				✓	✓	✓	4
4	✓	✓		✓				3
5	✓				✓			2
6	✓	✓		✓				**3**
7	✓		✓	✓				**3**
8	✓			✓			✓	**3**
9	✓							1
10	✓						✓	2
11							✓	1
; 12	✓		✓	✓				3
13	✓		✓	✓				**3**
14	✓							**1**
15						✓	✓	2
16	✓				✓	✓	✓	**4**
17	✓							1
18	✓		✓	✓	✓	✓		5
19	✓		✓	✓			✓	**4**
20	✓	✓						2
21	✓				✓		✓	3
Sum	17	3	5	9	5	4	7	

The number of concepts learned correlates positively, but only as a trend, with both percent correct ($\rho = .29$, $p < .10$) and number of moves ($\rho = .34$, $p < .07$). Percent correct and number of moves are not correlated ($\rho = -.22$, $p < .84$). In Figure 4 we show scatter plots of the pairwise comparisons between these variables.

For stimuli with a correct answer where the subject used the correct method, we had 470 satisfaction ratings with $\mu = 5.88, \sigma = 1.37$. For stimuli with a correct answer where the subject used the incorrect method, we had 444 satisfaction ratings with $\mu = 4.94, \sigma = 1.77$. A t-test indicated a significant $p < .01$ effect of correct versus incorrect method on satisfaction, indicating that subjects were in general able to judge some difference between good and bad partitions.

Discussion

Almost every subject learned about cluster analysis through their experience–over half learned three concepts or more. Giving researchers expertise and access through tools like Divvy promises to encourage and improve the application of machine learning techniques in other fields.

Nevertheless, some subjects had difficulty using the knowledge they acquired to make good data analysis decisions. Though subjects explored quite a bit during the training phase (an activity that showed a trending correlation with concept learning) they did not necessarily parlay that experience into better performance. So while we are pleased that subjects demonstrated concept learning in the interviews, we would like to investigate why they had trouble applying it. The subjects were overall less satisfied when using the incorrect method, which indicates that evaluative confusion was not the primary culprit.

Given that the core audience for Divvy is composed of graduate students, postdocs, and faculty, we would like to perform a follow-up study with that audience. While undergraduates serve as a useful lower bound, so to speak, for testing learning with Divvy, our target population is likely more motivated, more familiar with data analysis tasks, and in possession of greater domain knowledge.

The process of crystallizing the implicit knowledge gained during the experiment in the interview might help subjects make better decisions. To test this, a future experiment could place the interview between the training and test blocks. If this results in better performance, it would indicate that having to articulate knowledge assists concept crystallization and application, and that the subjects are in a sense still learning when they fill out the interview.

We do not think a comparison to traditional guided learning is useful since our target population will rarely have the time or inclination to seek out explicit instruction. However, we would be interested in comparing our results to other forms of discovery learning where the interaction between subject and sofware is modified. We believe that self-directed exploration with instantaneous feedback is valuable and we would like to compare our results with, e.g., simply showing subjects a set of partitions and their associated methods and parameter values without allowing them to choose parameters, or putting a delay between parameter changes and result visualization. These modifications would move the experimental context closer to traditional machine learning approaches where the training data are fixed (as opposed to the active learning paradigm mentioned earlier). It would also correspond to writing a script to run through a set of parameter settings and visualizations while one goes out for coffee, and then interpreting when one returns.

Our results provide compelling evidence that undergradu-

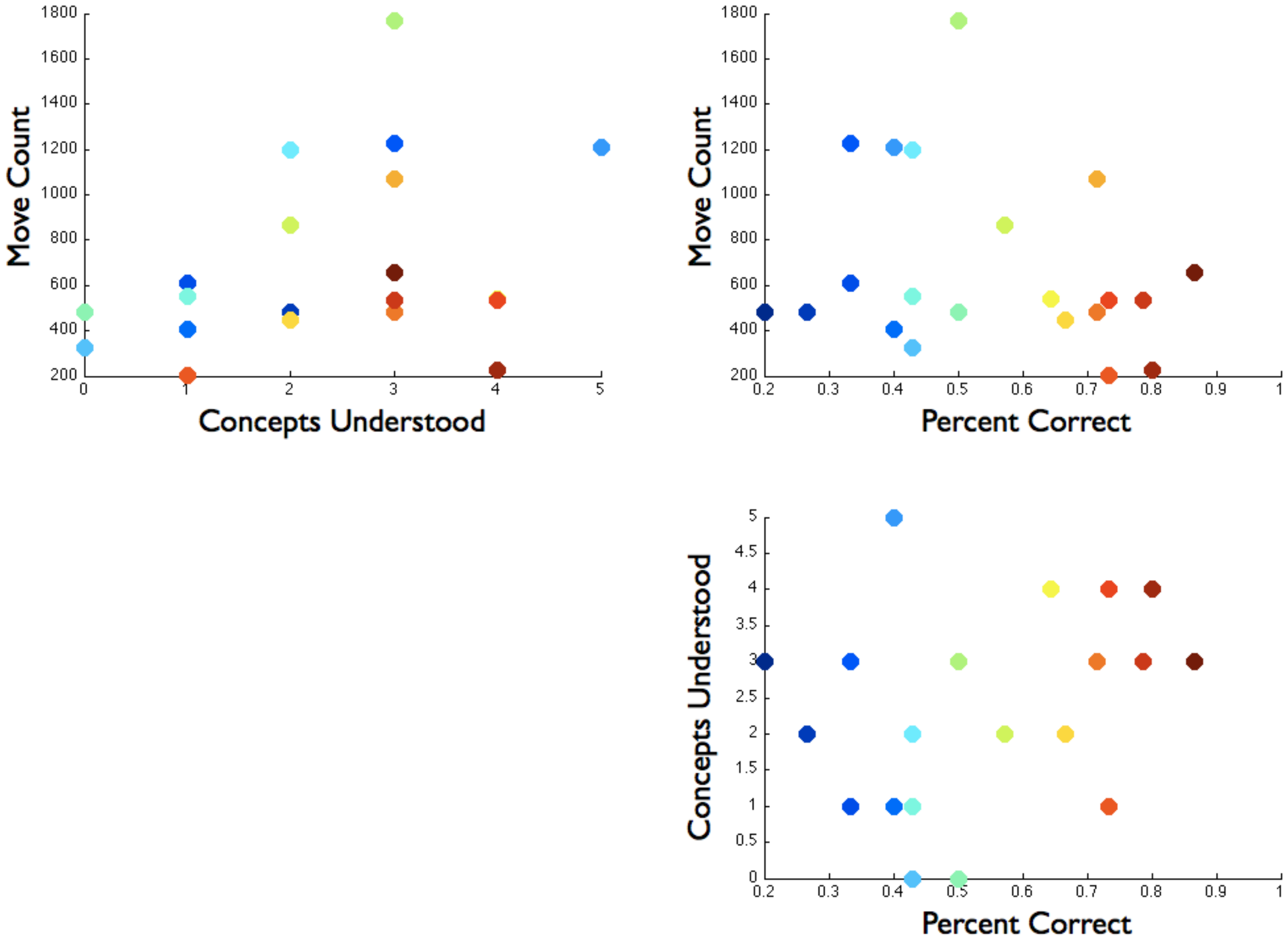


Figure 4: Scatter plots of the three main variables. The points are colored from dark blue to dark red based on percent correct.

ate subjects can learn useful concepts about machine learning algorithms just by interacting with them. This leads one to suspect that the target population for this work, practicing researchers, will be able to do so as well. Subjects do not reliably apply these concepts when tested, and additional study is required to determine why this is, and how to better support the discovery and application of machine learning concepts.

Acknowledgments

This work is funded by NSF Grant #SES-0963071, Divvy: Robust and Interactive Cluster Analysis (PI Virginia de Sa). Thanks to Cindy Zhang for valuable code contributions.

References

Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review*, *31*, 21–32.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *CoRR*, *cs.AI/9603104*.

Ggobi data visualization system. (n.d.). http://www.ggobi.org.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, *41*(2), 75-86.

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations.* Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66, 1, 281-297.

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? the case for guided methods of instruction. *American Psychologist*, *59*, 14–19.

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning* (pp. 873–880). New York, NY, USA: ACM.