Regina Eckert, Grace Kuo
9/29/17
Machine Learning in Education

Visualizing representations of problems and skills

1. Train a genism skip-gram model of the skills.tsv from the last assignment.

```
#Read out the "sentences"
sentences=skill_df.iloc[:,1:].values.astype(str)
sentences=sentences.tolist()
#Each student is a "sentence", each skill is a "word"
#size = dimensionality of feature vectors
#window = max distance between current and predicted word within
a sentence
#min_count = minimum number of occurrences within dataset
#workers = number of threads used
#sg = 0 (CBOW, default); = 1 (skip-gram)
model = Word2Vec(sentences, size=100, window=5, min_count=10,
workers=4, sg=1, iter=30)
```

2. Use t-sne to reduce dimensionality.

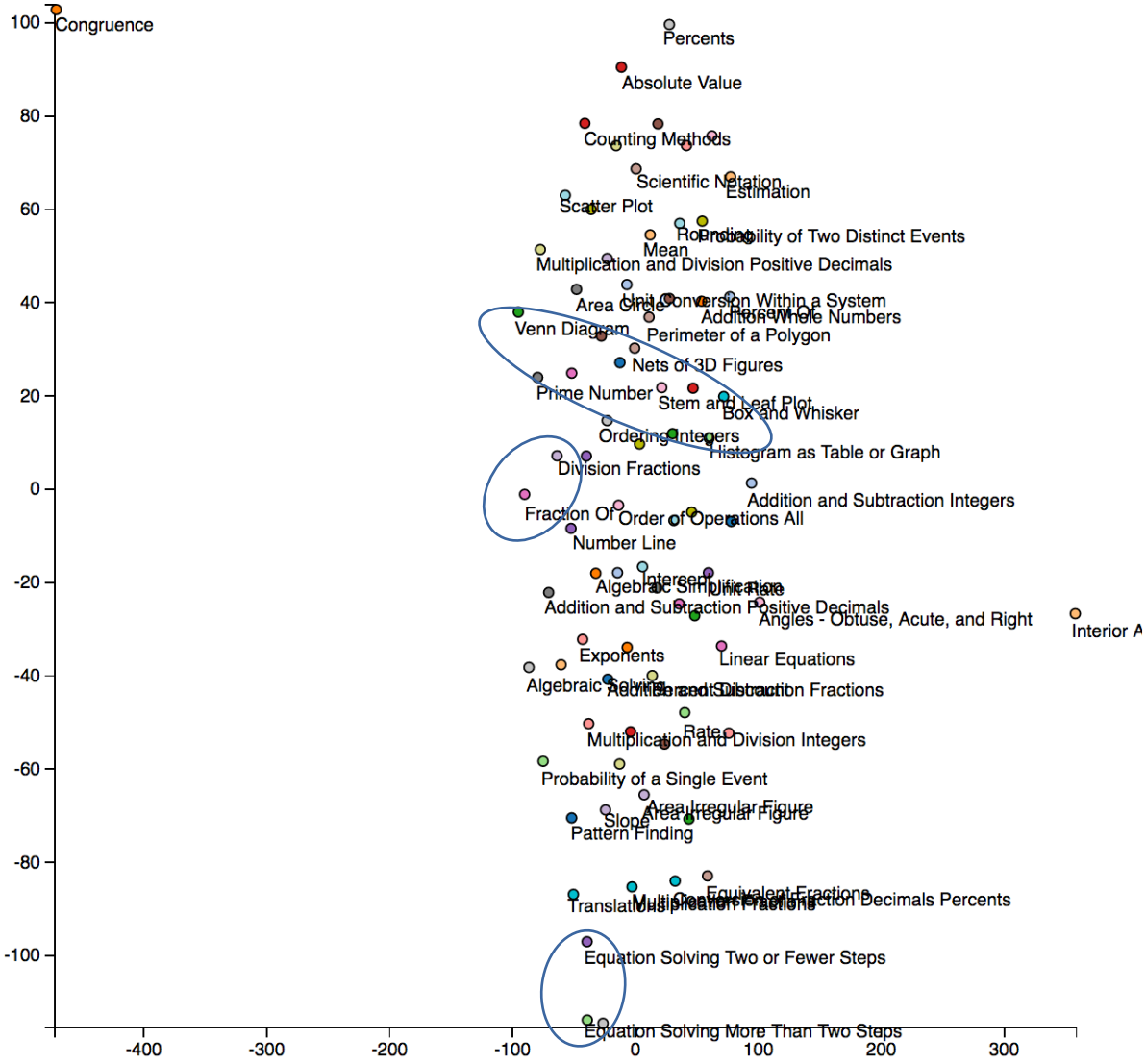
```
skill_num=model.wv.vocab; #Names of the words (numbers)
skill_vec=model[skill_num] #Access the vectors
tsne=TSNE(perplexity=30) #Instantiate the TSNE model (can change
params here)
skill_tsne=tsne.fit_transform(skill_vec) #Run tsne
```

3. What do you make of the structure of the visualization? Is there a logic to the proximity of skills to one another?

Word2Vec: size=100, window=5, min_count=10, workers=4, sg=1, iter=30
T-Sne: perplexity = 30

While the structure of the graph is not very clustered (with just one main cluster and a few outliers), by inspecting the names of the skills we can find some groups of skills that make sense. These include:

- Plot types: box & whisker, stem & leaf, venn diagram
- Group of fraction-related topics
- Equation solving



4. Re-train the skip-gram model using different hyperparameters (try a window size of 1). How does the vector size and window size appear to affect the visualization?

We changed a variety of hyperparameters for the skip-gram (changing the feature vector size, the window, and the number of iterations) as well as trying to tune the perplexity of t-sne.

Generally, we found that scaling the feature vector size and window size together generated more meaningful graphs. We theorize that the relationships are more simple when just considering the nearest neighbors (window = 1) and so need fewer features to describe the data (size=50), whereas expanding the context (window = 10) required more features for good description (size=200). That being said, a feature vector size 25 was too low to give any meaningful results regardless of window size.

However, we also found that the “perplexity” of the t-sne reduction made a huge difference in the understandability of the graph relations. For example, with (size = 50, window =1, and iter=100), we get a much more clustered, meaningful graph with perplexity =10 rather than perplexity = 30. However, with (size = 200, window =5, and iter=100), the opposite is true.

Overall, we think this data would be easier to visualize by defining connections between nearest neighbors (say, the 5 most similar vectors are connected) and then doing some graph-based projection to 2D. This way you could see which skills were connected, because the t-sne optimization has many local minima that make it hard to discern underlying connections for this data.

Vector Size	Window	# Iter	T-Sne Perplexity	Observations
100	5	30	30	Not clustered, but can observe groups: - Plot types: box & whisker, stem & leaf, venn diagram - Group of fraction-related topics - Equation solving
50	1	30	30	Less meaningful clustering – hard to pick out groups Spread out graph
50	1	100	30	Lots of spreading, but can still discern groups such as: - Plot types (box & whisker, stem & leaf, circle graph, histogram as table) - Fractions - Linear equations - Ordering
50	1	100	10	More clustered groups, less random spreading. Groups: - Fractions - Angles - Equation solving - Plot types
50	2	100	30	Starting to get spreading again, but can see groups: - Plots - Some fractions (though some spread out) - Mean, median, mode - Area
50	4	100	30	Very little meaningful clustering or grouping
50	5	30	30	Very little meaningful clustering or grouping
50	10	100	30	Very little meaningful clustering or grouping
25	5	100	30	Extremely little meaningful clustering or grouping Data is mostly clustered in one close blob in the

				center of the graph, and seems randomly mixed in that blob
200	5	100	30	Fairly spread out graph, but can still pick out groups: - Ordering - Plot types - Solving systems of equations
200	5	100	10	Looks more clustered, and some small groups are discernable: - Area & shapes - Some fractions
200	1	100	10	Fairly spread out and seemingly mixed, with a few groupings: - Plot types (box & whisker, circle graph, number line)
200	10	100	30	Still fairly spread, but can see many groups: - Angle - Area - Plot types (box & whisker, stem & leaf, venn diagram) - Equations - Order of operations

Word2Vec: size=50, window=1, min_count=10, workers=4, sg=1, iter=30
T-Sne: perplexity = 30

- Less meaningful clustering – hard to pick out groups

Word2Vec: size=50, window=1, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 30

Lots of spreading, but can still discern groups such as:

- Plot types (box & whisker, stem & leaf, circle graph, histogram as table)
- Fractions
- Linear equations
- Ordering

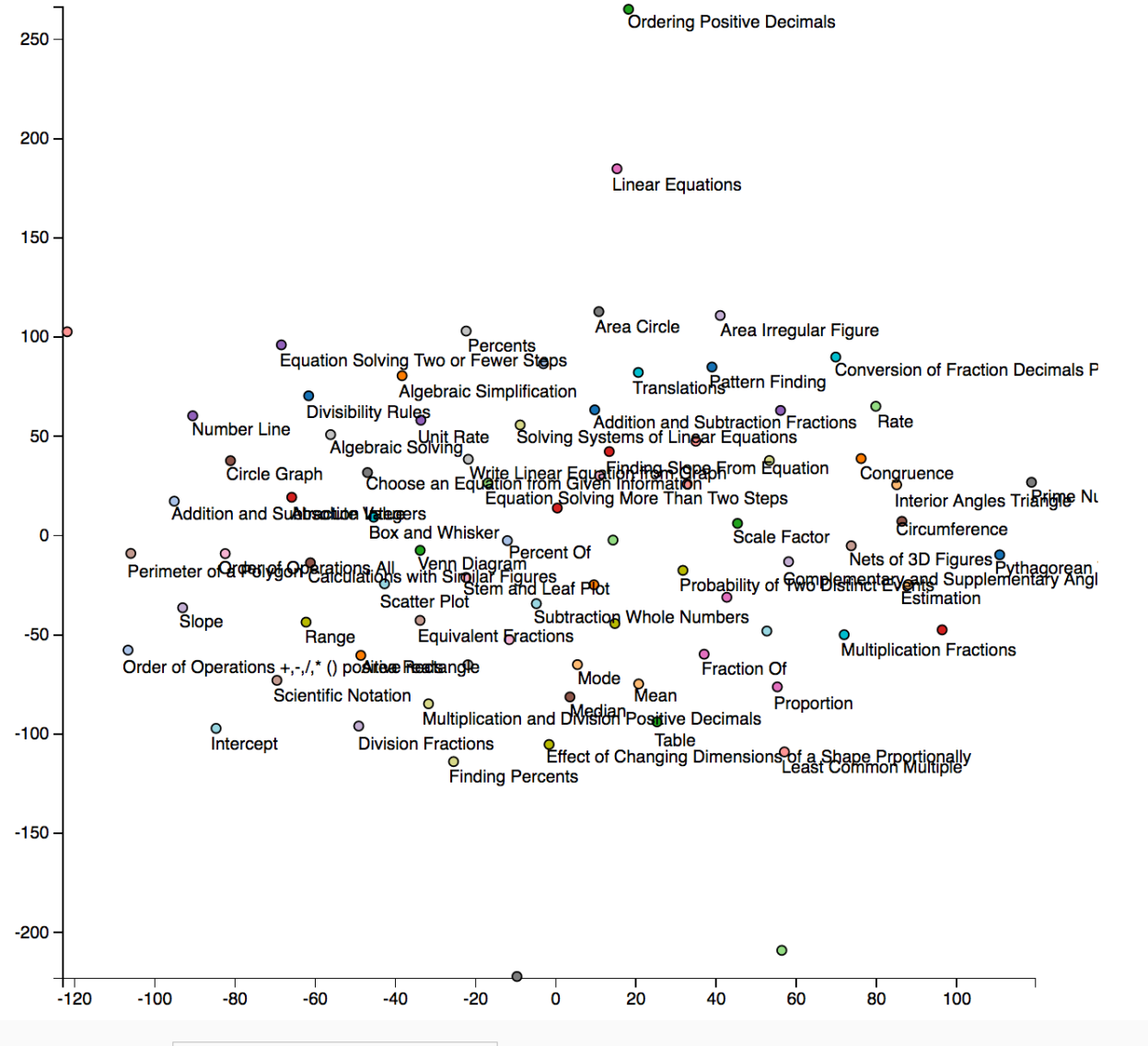
Word2Vec: size=50, window=1, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 10

More clustered groups:

- Fractions (top)
- Angles
- Equation solving
- Plot types

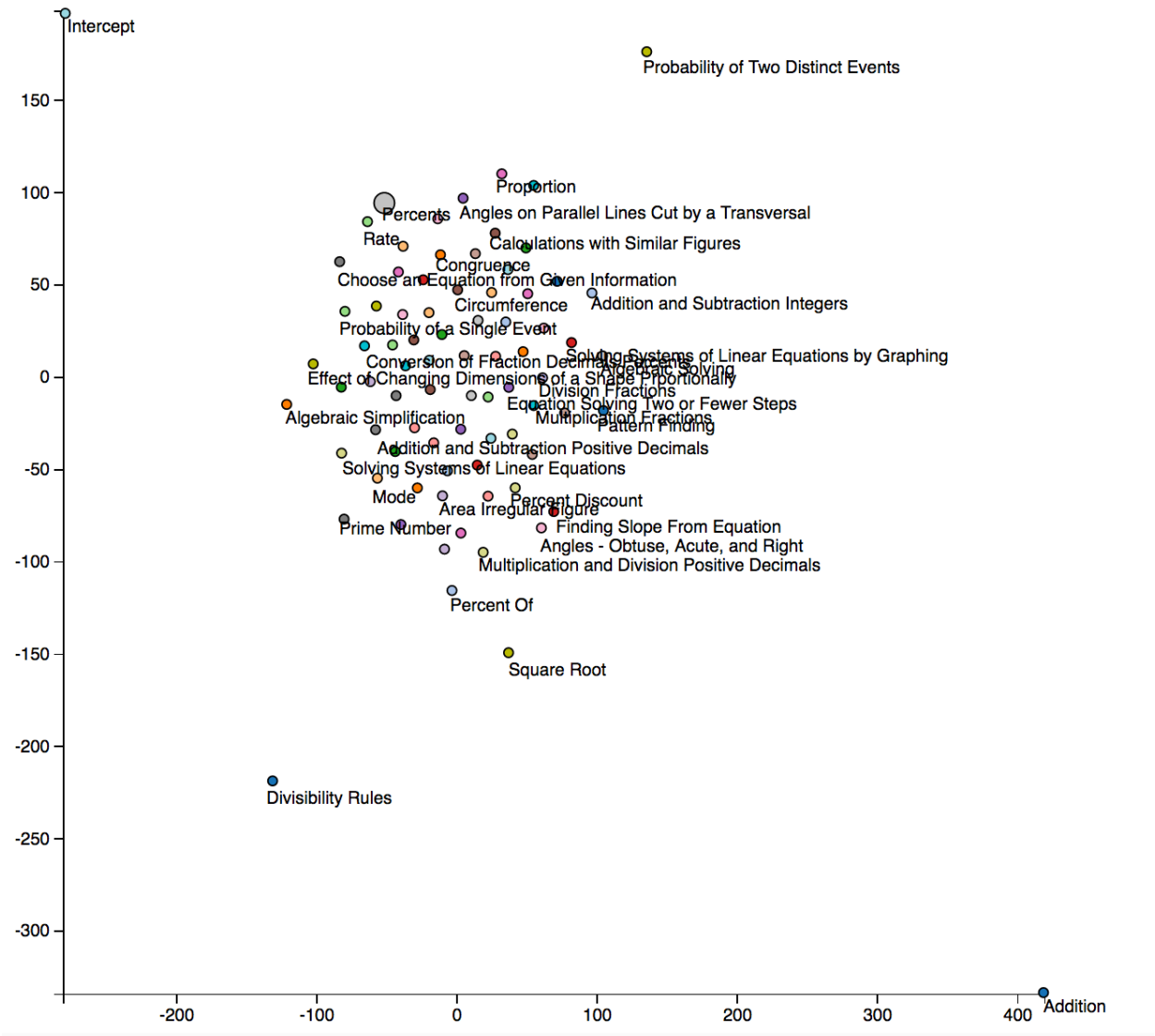
Word2Vec: size=50, window=2, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 30

Starting to get spreading again, but can see groups:
- Plots
- Some fractions (though some spread out)
- Mean, median, mode
- Area



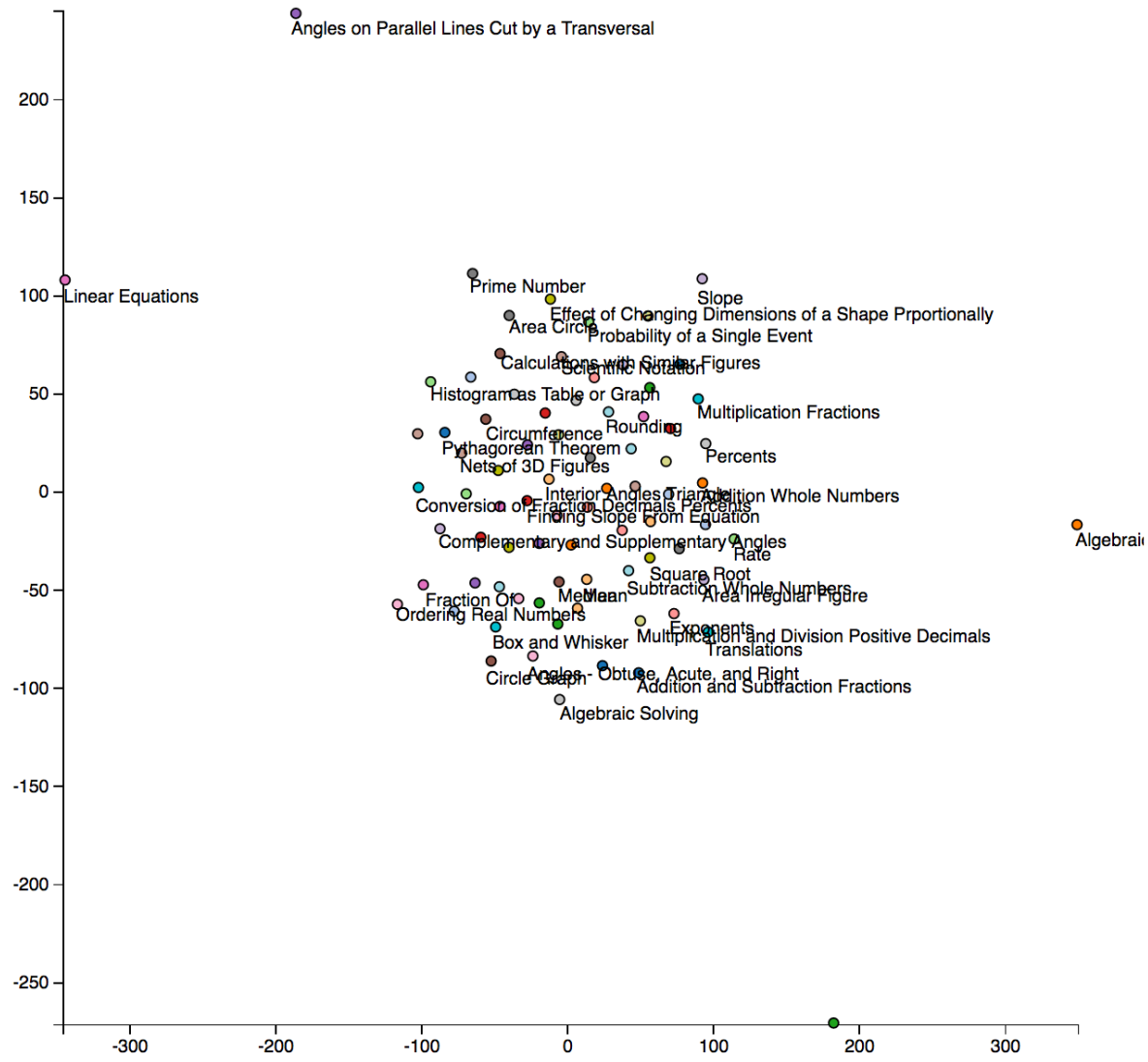
Word2Vec: size=50, window=4, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 30

- Very little meaningful clustering



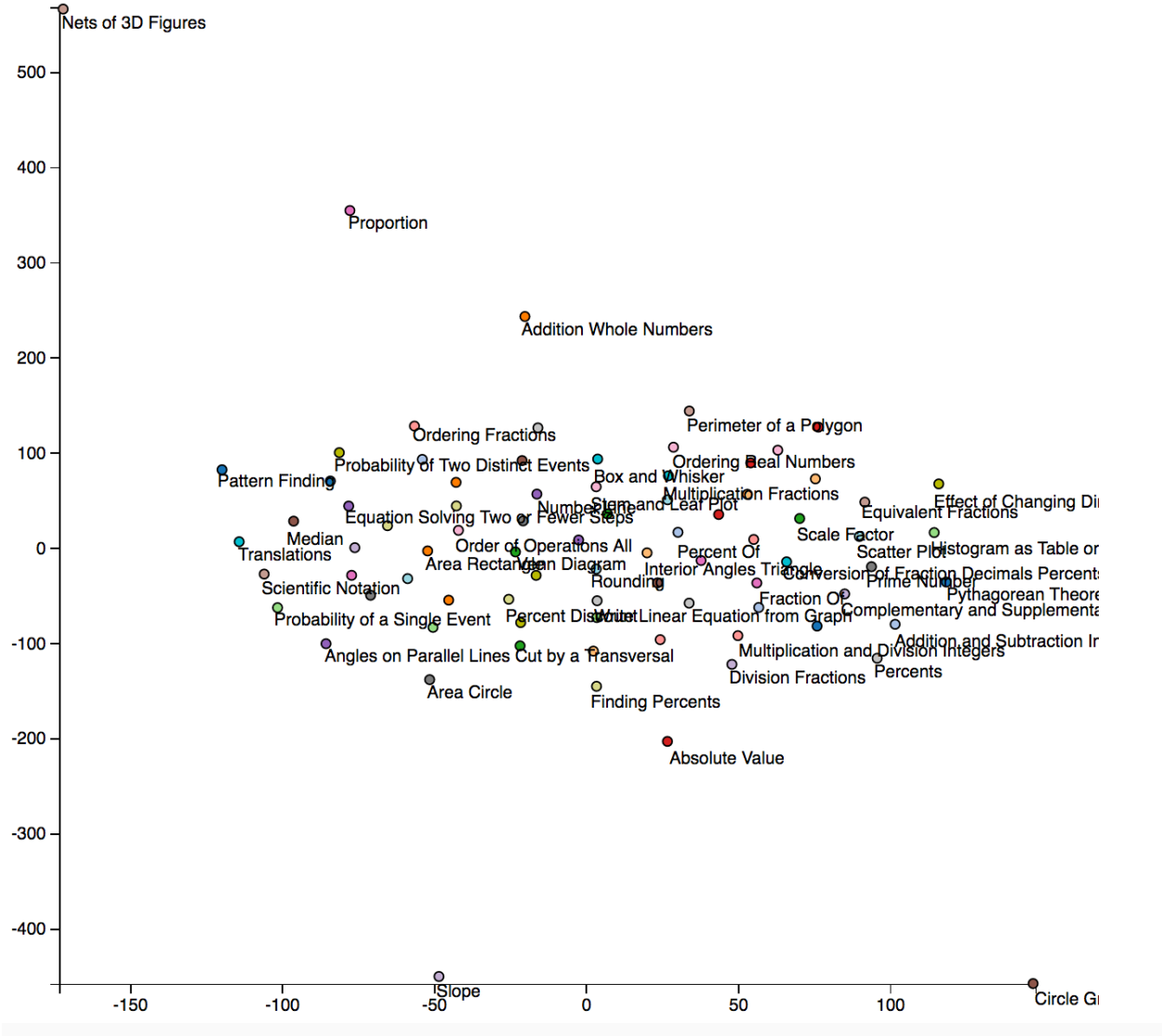
Word2Vec: size=50, window=5, min_count=10, workers=4, sg=1, iter=30
T-Sne: perplexity = 30

- Less meaningful clustering – hard to pick out groups



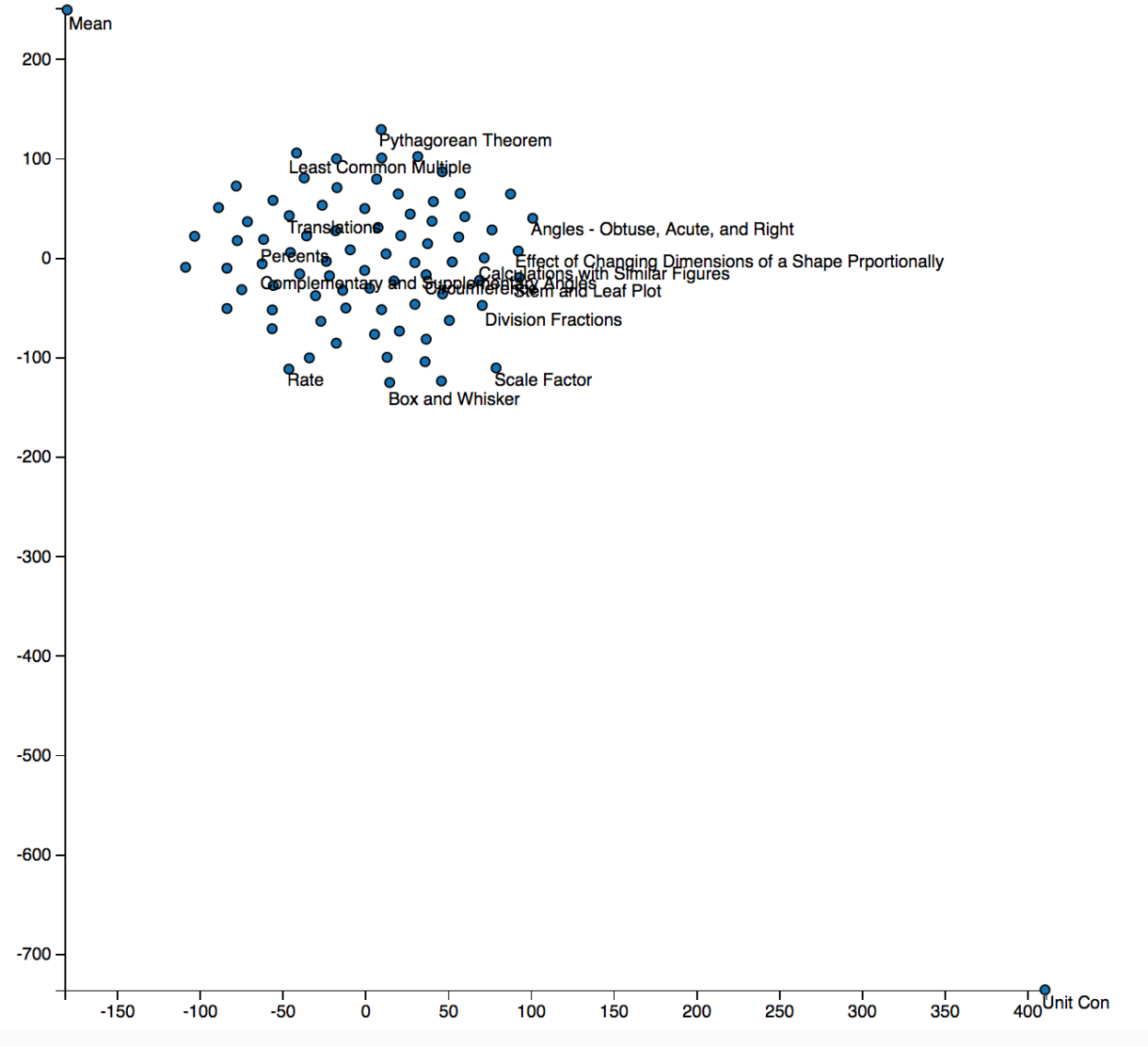
Word2Vec: size=50, window=10, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 30

- Very little meaningful info



Word2Vec: size=25, window=5, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 10

- Extremely little meaningful info



Word2Vec: size=200, window=5, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 30

Fairly spread out graph, but can still pick out groups:

- Ordering
- Plot types
- Solving systems of equations

Word2Vec: size=200, window=5, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 10

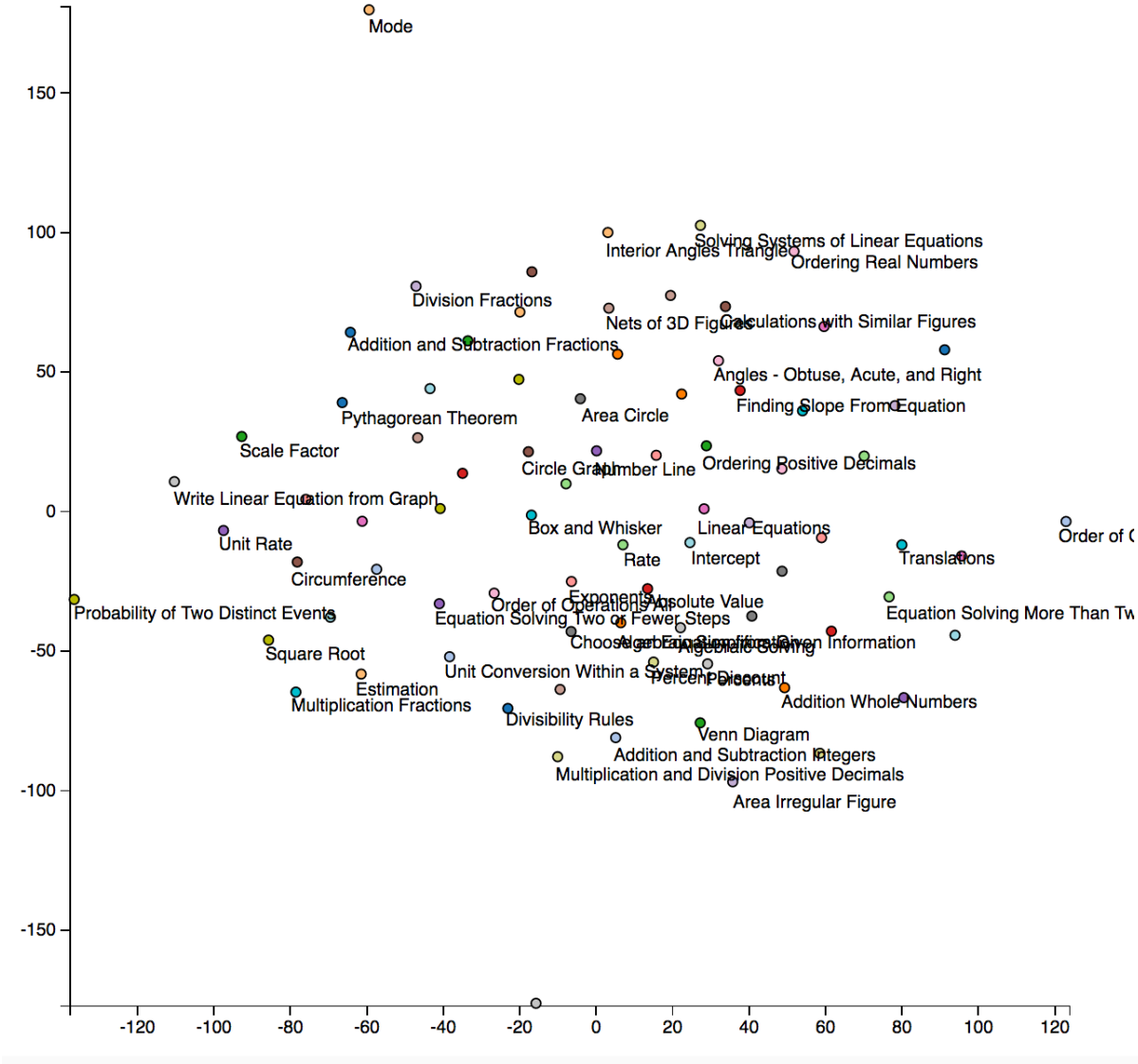
Looks more clustered, and some small groups are discernable:

- Area & shapes
- Some fractions

Word2Vec: size=200, window=1, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 10

Fairly spread out and seemingly mixed, with a few groupings:

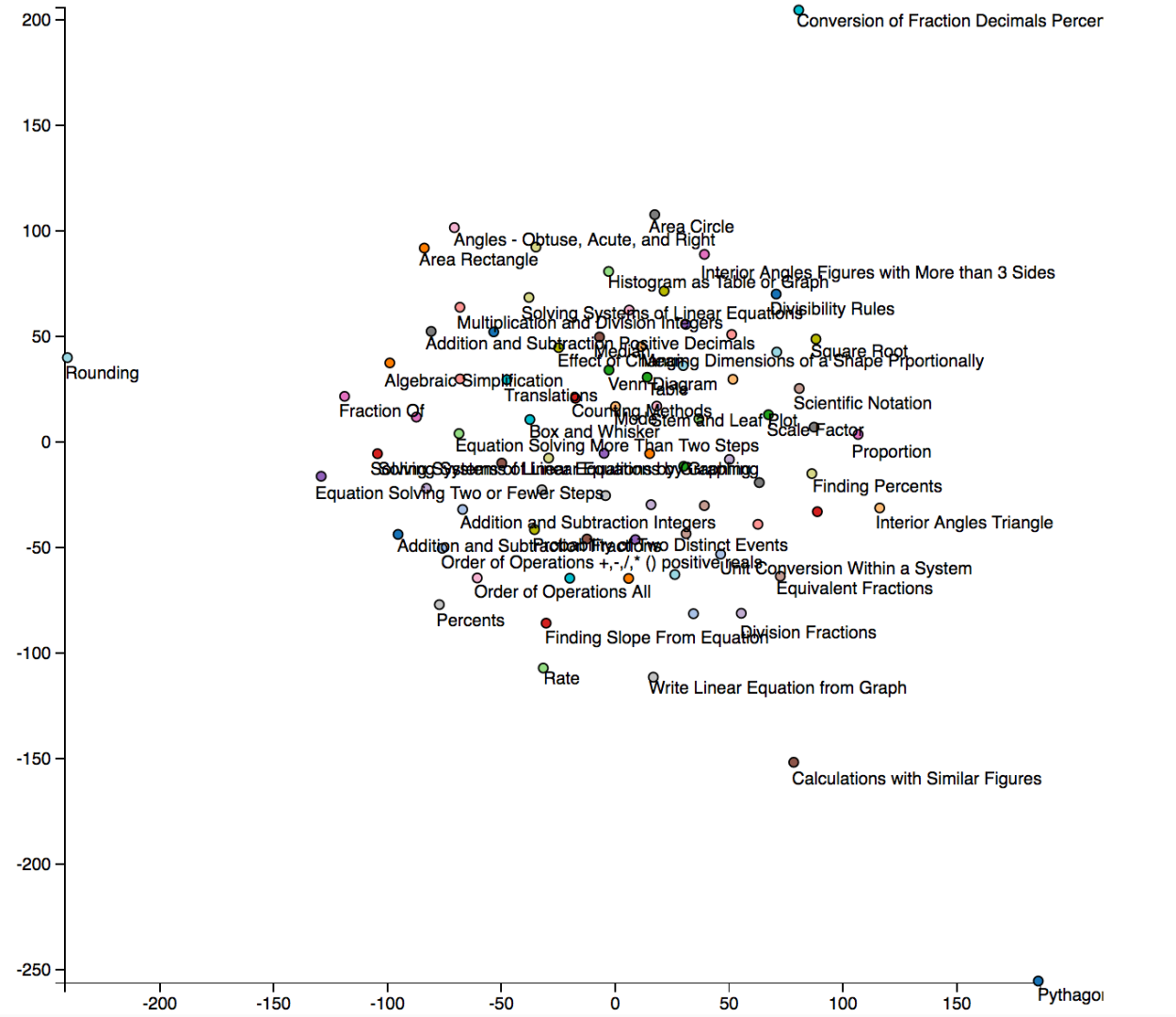
- Plot types (box & whisker, circle graph, number line)



Word2Vec: size=200, window=10, min_count=10, workers=4, sg=1, iter=100
T-Sne: perplexity = 30

Still fairly spread, but can see many groups:

- Angle
- Area
- Plot types (box & whisker, stem & leaf, venn diagram)
- Equations
- Order of operations

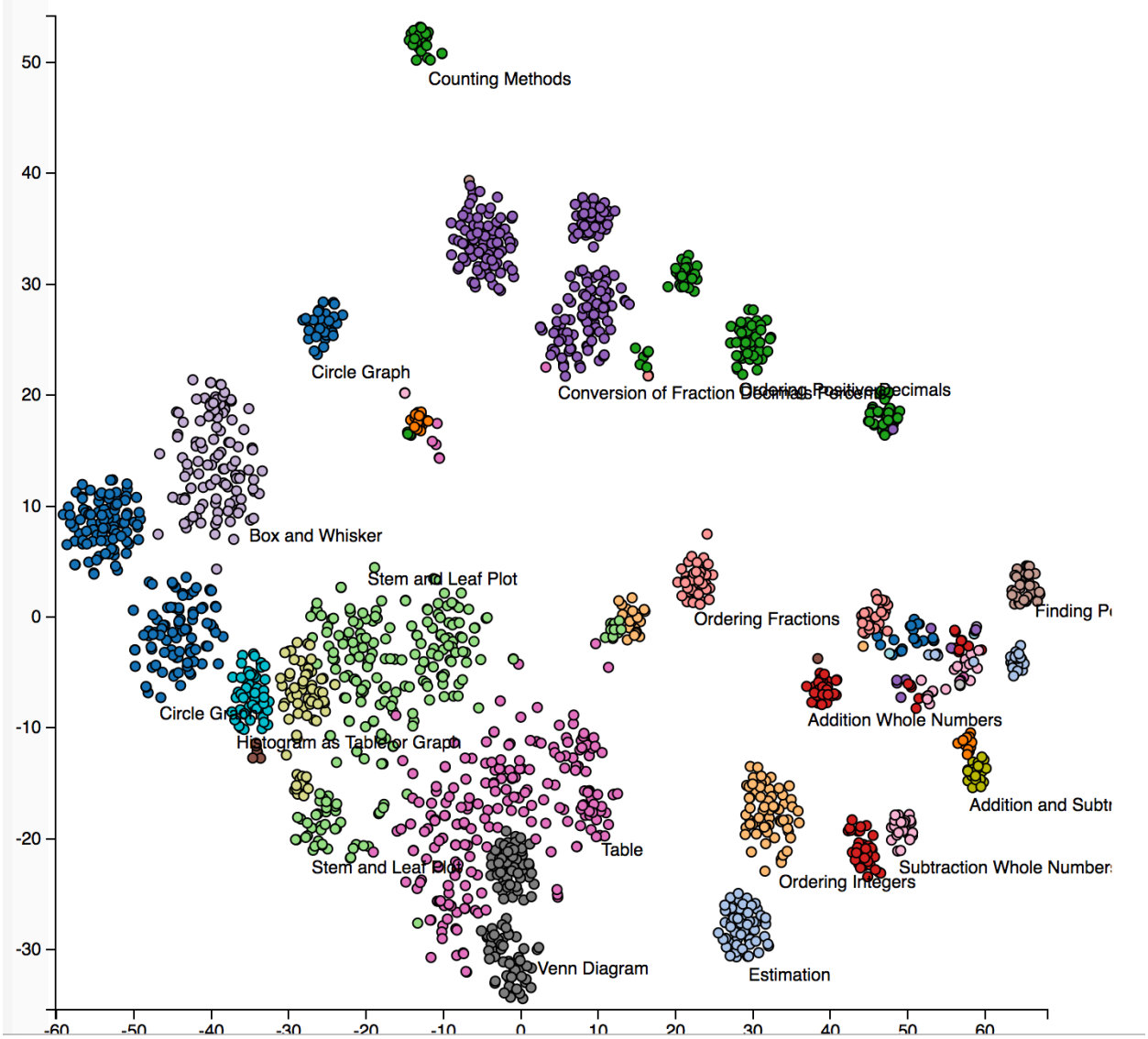


5. Repeat steps 1 through 4 but using the assistments_id.tsv file instead. This will be a visualization of problems (more numerous) instead of skills.
   Color assistment plot point by their skill. Do they cluster by skill?

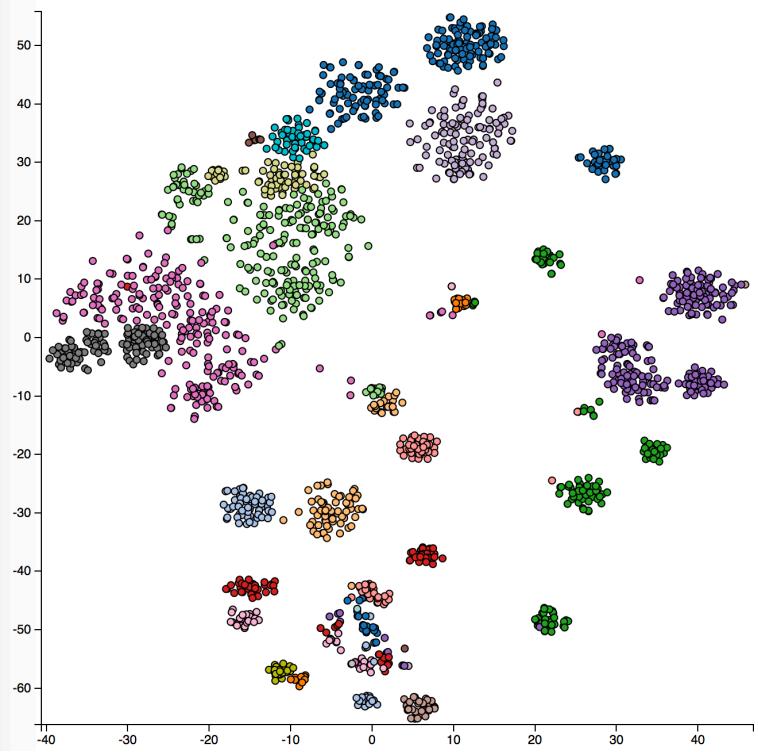
Below are visualizations of the Assistment Ids, colored by the first skill associated with them in the data set (many were associated with multiple skills). There is a lot of clear clustering of skills, which is neat. We varied the feature dimensionality, setting it at 200, 100, 50, and 25. This did not effect on the visualization much, perhaps because of the dimensionality reduction.

It looks the data-visualization assignments (circle graph, venn diagram, box and whisker, stem and leaf, table) are grouped together, and so are arithmetic assignments (Ordering integers, addition whole numbers, addition and subtraction, subtraction whole numbers).

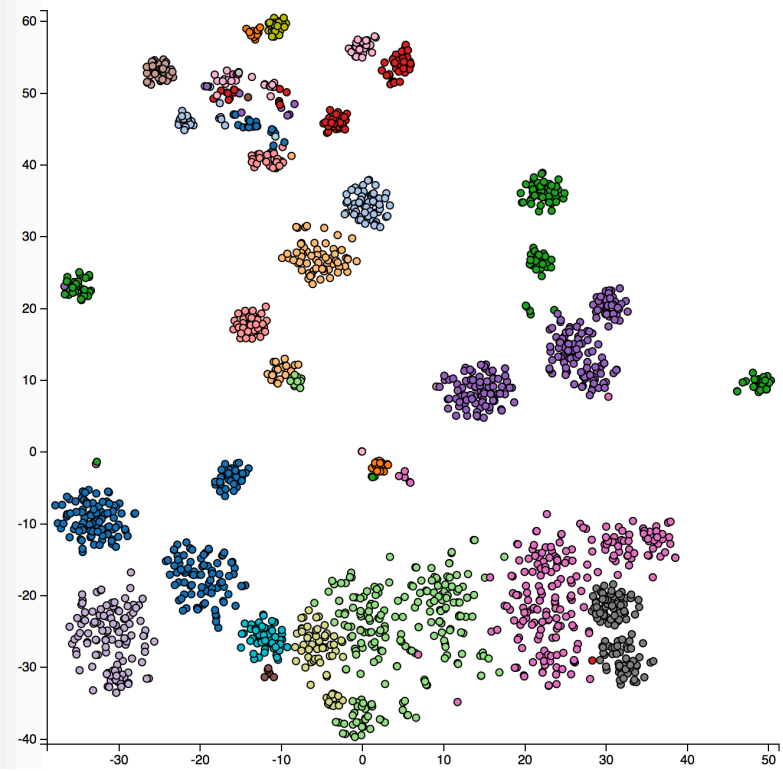
Feature Dimensionality = 200



Feature Dimensionality = 100



Feature Dimensionality = 50



Feature Dimensionality = 25