Machine Learning & Data Mining
Miniproject 2

Caltech CS/CNS/EE 155
February 2021.

---

1 Introduction [5 points]

- Group members: Francesca-Zhoufan "plotting perfectionist" Li, Elena Sorina "surprise master" Lupu, Nikhil "movie expert" Ranganathan
- Team name: Pandas
- Work division: Francesca and Nikhil worked on the first part to get basic visualizations. Sorina, Francesca, and Nikhil worked each on a method from the 3 methods and generated corresponding plots. Nikhil chose interesting movies and provided insights. Sorina troubleshot surprise. Francesca performed nevergrad optimization. All worked on the report and on the google colab.
- Packages used: Visualization: matplotlib, seaborn, bokeh, holoviews, iqplot; Modelling: surprise, sklearn; Optimization: nevergrad
- Note: since there is no page limit and the fact that visualization and data analysis can be quite tailored to personal preferences, we thought it is the best to be thorough and holistic so we hope you would enjoy :)

2 Basic Visualizations [20 points]

We noted that there are some entries with everything the same but the MOVIE_ID, indicating duplicated data. There are also cases that the same user rated the same movie multiple times with different values, we take the average of such rating. With the philosophy of "plotting all your data" (credit to Dr. Justin Bois and BEBi103), we tried histogram over the selected group of interest, histogram of each movie, ECDF (empirical Cumulative Distribution Function), strip-box plots, violin-box plots, heatmap. We used histograms and violin plots with some caution as histograms need bins and violin plots can have kernel densities that are not as exact as how ECDFs, strip plots, and heatmaps can represent all data. However, considering that ECDF might not be the most intuitive and prevalent for many, we included other forms of visualizations. Also just to note that in the case of histogram where all ratings are integers, it would be fine either with ECDF of histograms but there are some as mentioned above (an average of multiple ratings) which might not necessarily be integers in which case, ECDF would be a more accurate representation. One might argue that heatmaps are not the best as the x and y axis would not be as meaningful. However, we do think it can be a quick and dirty way of getting a general sense of how many genres are combined with each other and how some movies have a lot more ratings than the other though sometimes the plots can get quite crowded. Given the philosophy also from Justin, that interactive plots are the way to go, we would let the readers to play with some of the plots or have closer examination of the plots that cannot be included here.

Genre Summary

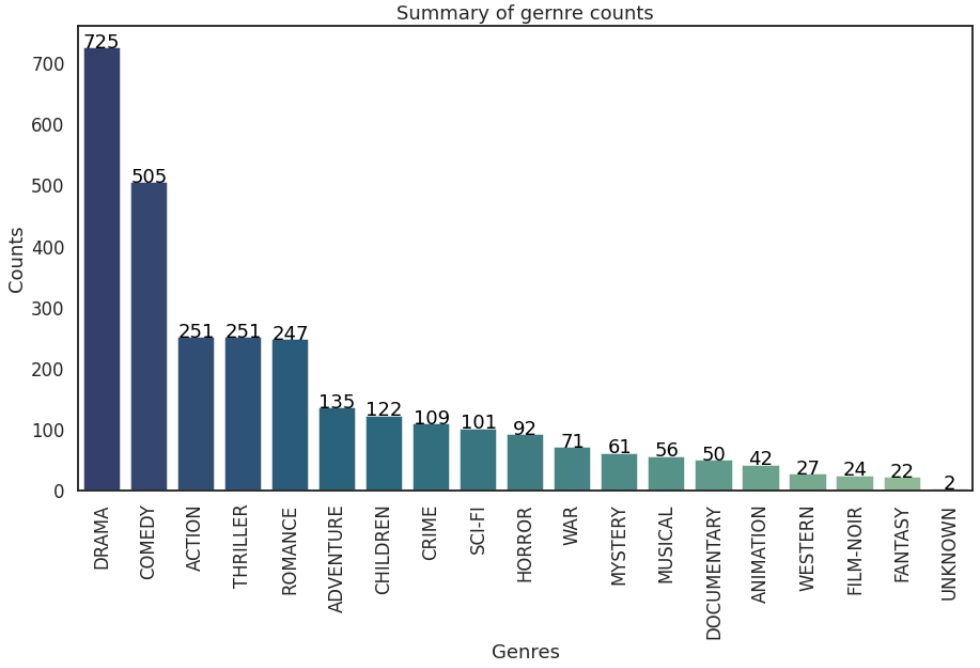

Figure 1: Summary of all genres in MovieLens dataset

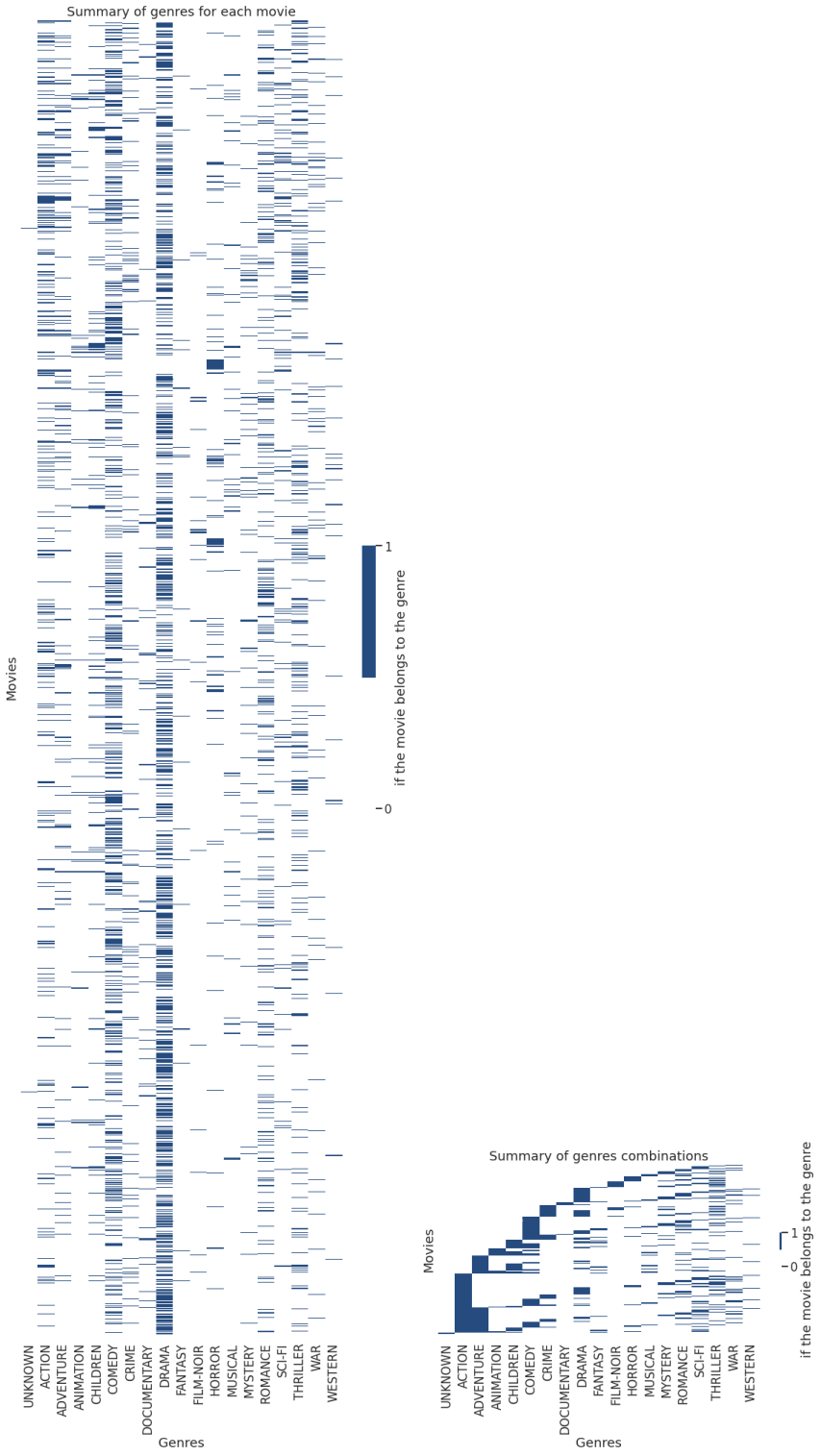

Figure 2: Each movie has multiple genres and there are multiple genre combinations in MovieLens dataset

All ratings in the MovieLens Dataset

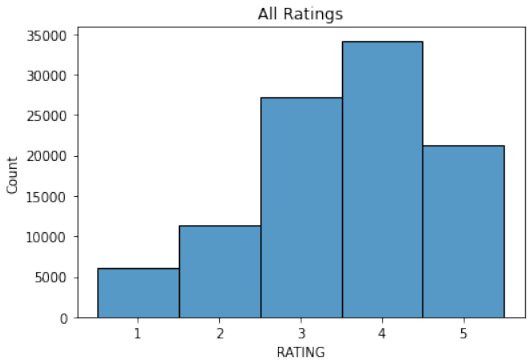

Figure 3: Histogram of all ratings in MovieLens dataset

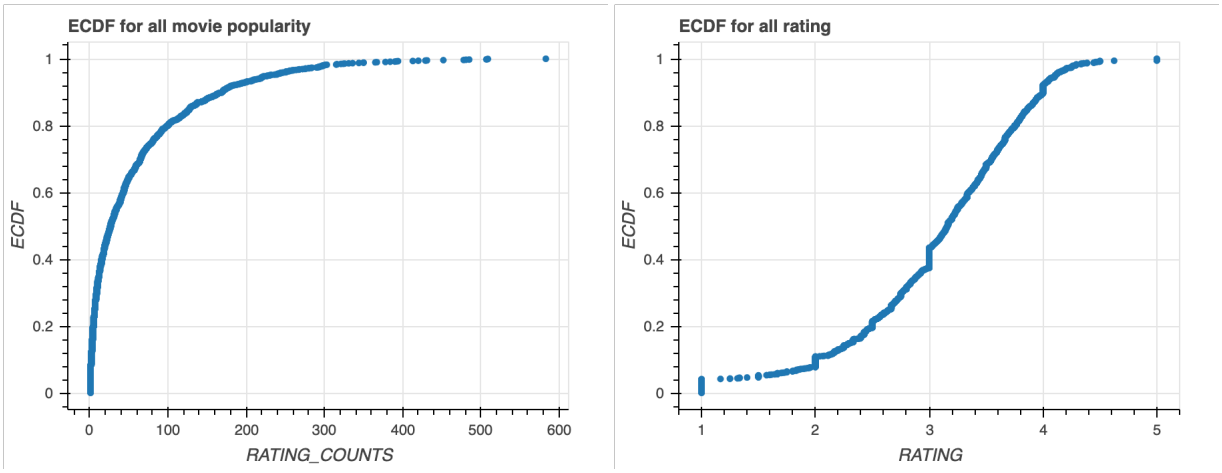

Figure 4: ECDFs of all rating counts and avreage ratings in MovieLens dataset

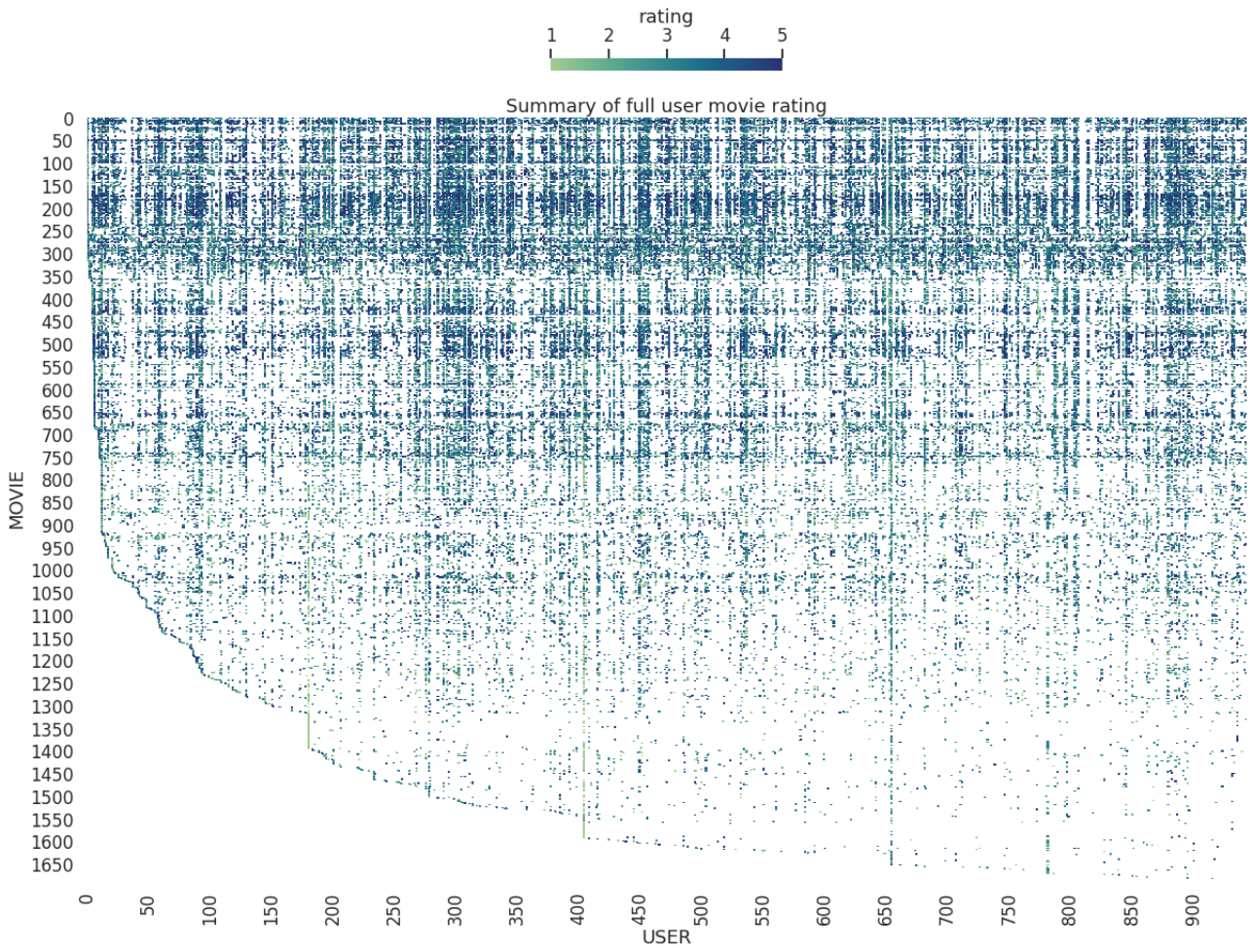

Figure 5: All movie rating in MovieLens dataset

Ten most popular movies

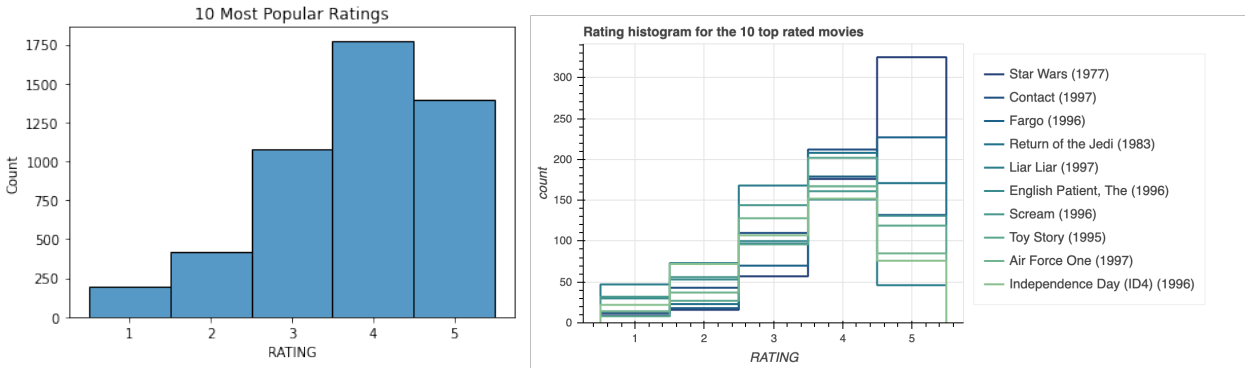

Figure 6: Histogram of 10 most popular movies with clickable legends in Colab

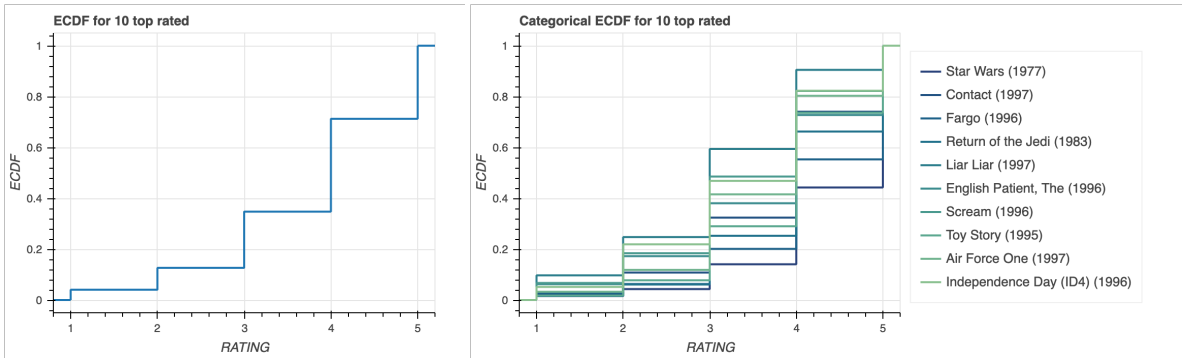

Figure 7: ECDFs of 10 most popular movies with clickable legends in Colab

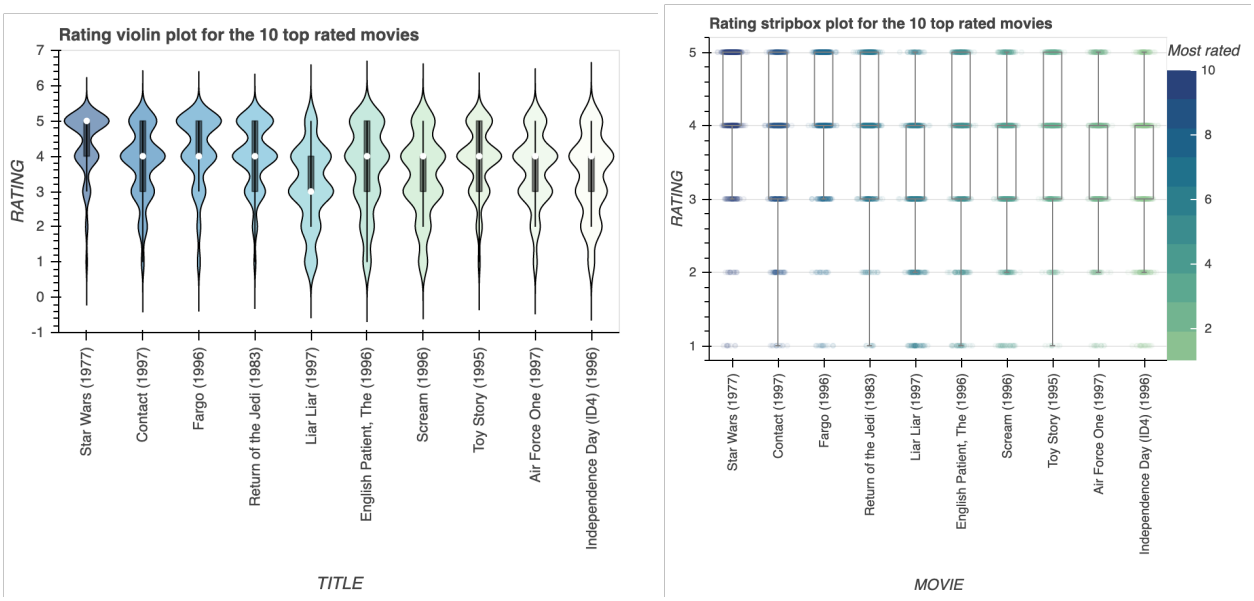

Figure 8: Violin plots and stripbox of 10 most popular movies

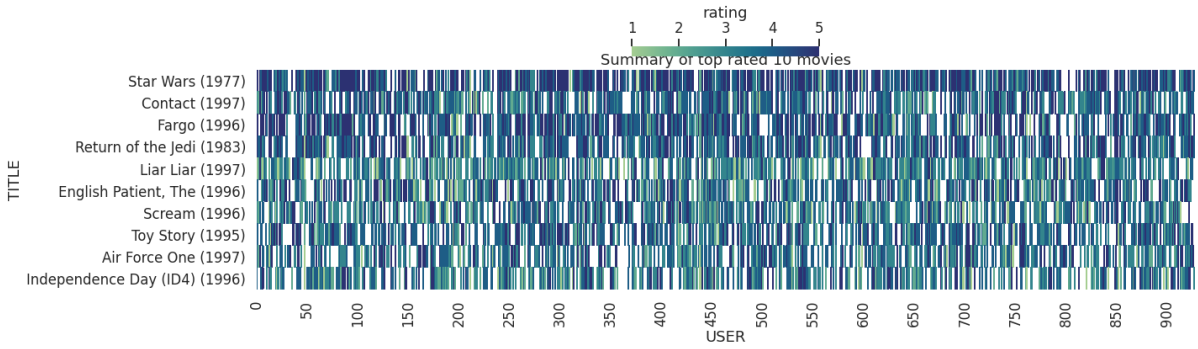

Figure 9: Heatmap rating of 10 most popular movies

Ten highest rated movies

Note that the highest rated ones can have as few as one or two ratings which is not representative nor interesting.(See Colab) At the same time, we also do not want to cut too many movies off. Cutting at 5 would leave us with 80% of the data, which is reasonable.

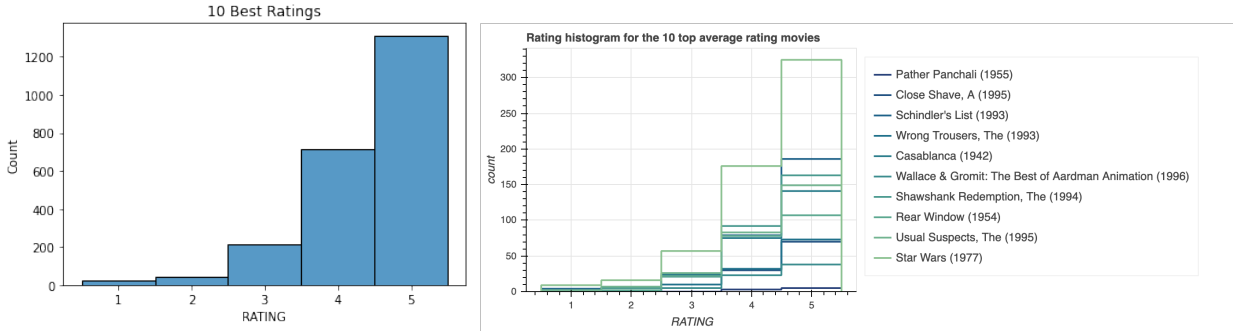

Figure 10: Histogram of 10 most popular movies with clickable legends in Colab

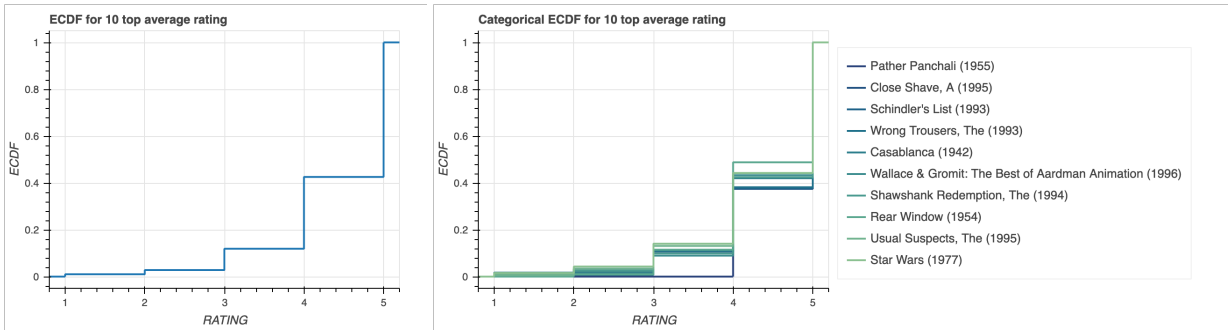

Figure 11: ECDFs of 10 most popular movies with clickable legends in Colab

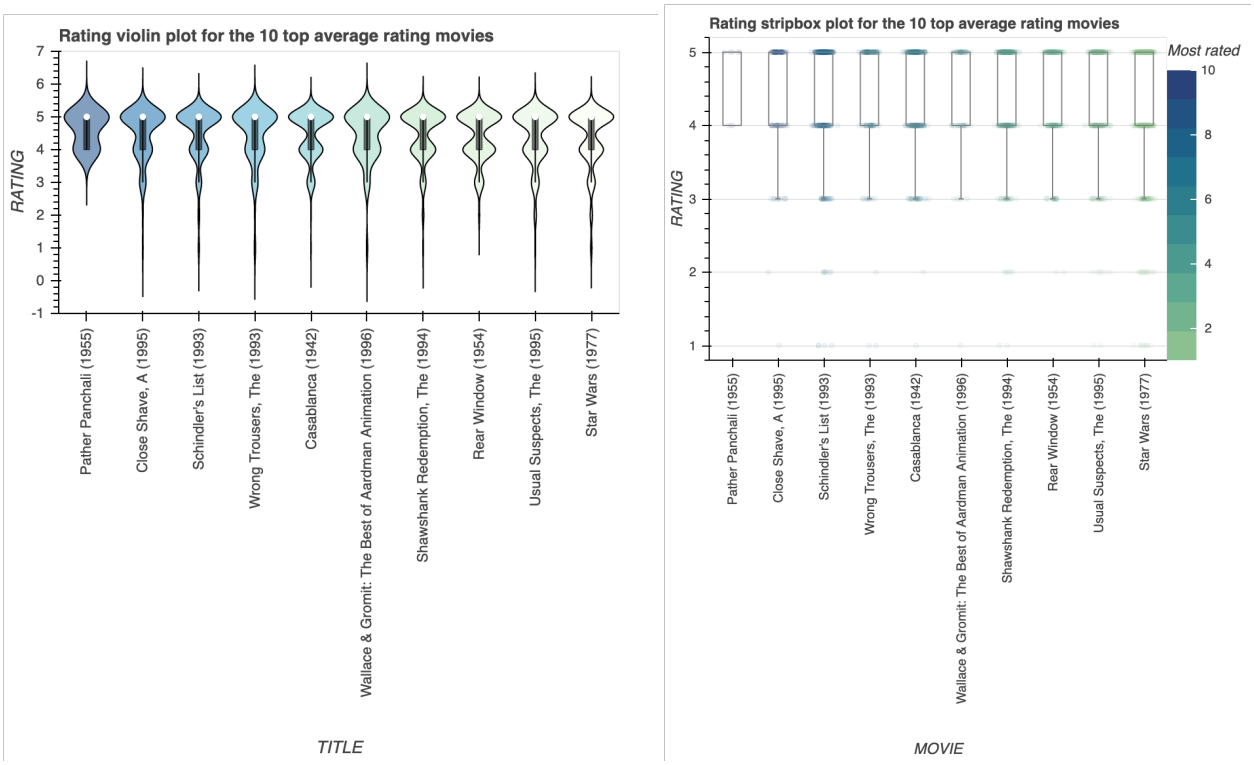

Figure 12: Violin plots and stripbox of 10 most popular movies

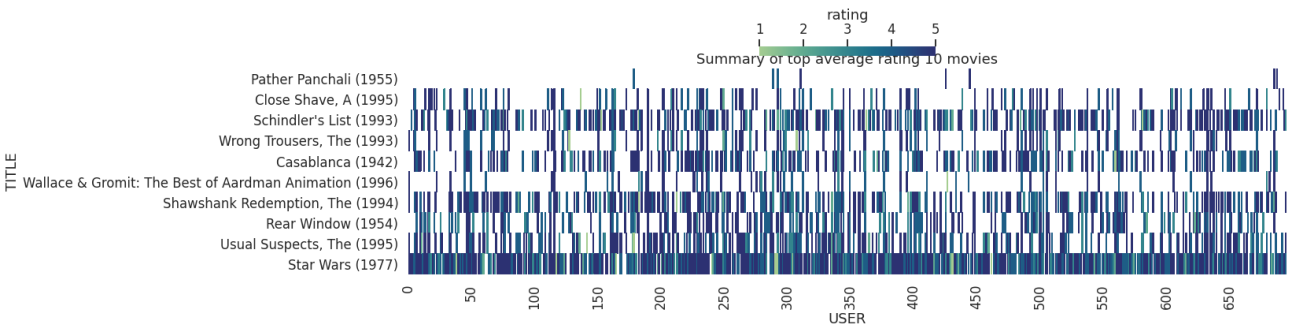

Figure 13: Heatmap rating of 10 most popular movies

Chosen genres

We chose comedy, sci-fi, and romance as three genres. Due space restrictions, it would not be as aesthetic and readable to have overlays, and thus we present histogram and the ECDF summary plots here. The rest can be found in the Colab notebook.

Chosen genre 1: Comedy

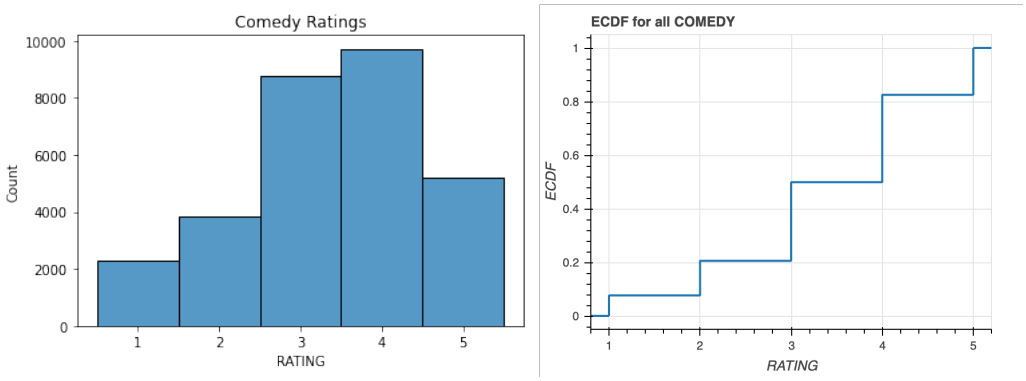

Figure 14: Histogram and ECDF of ratings of comedy movies

Chosen genre 2: Sci-Fi

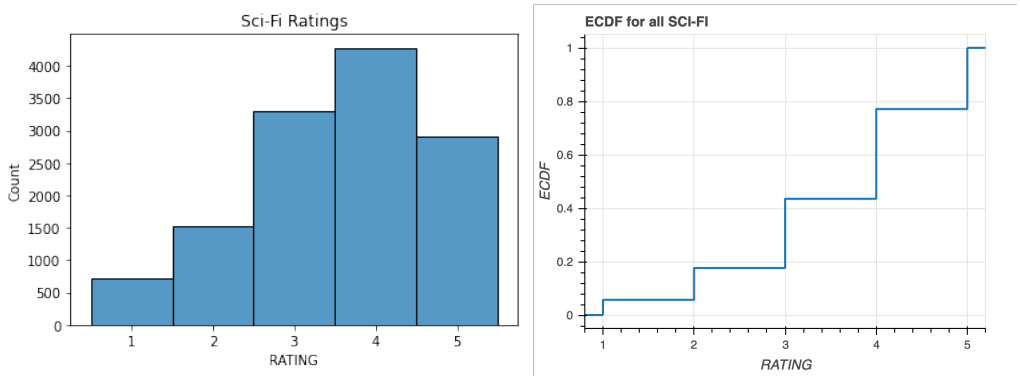

Figure 15: Histogram of ratings of sci-fi movies

Chosen genre 3: Romance

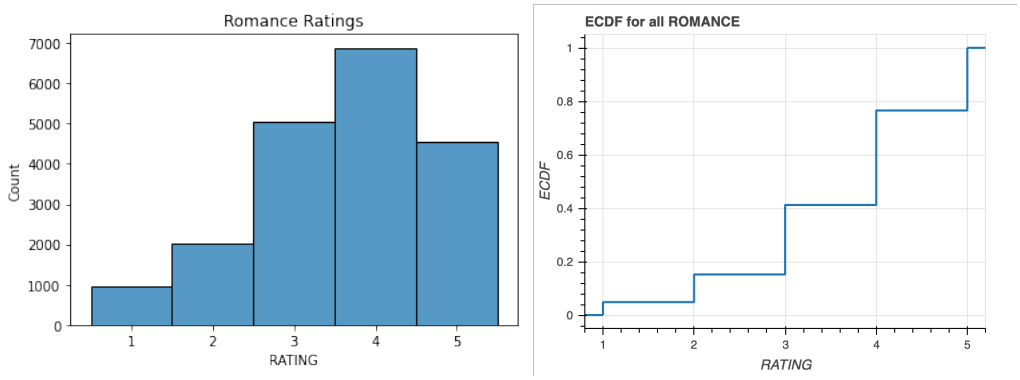

Figure 16: Histogram of ratings of romance movies

Discussion

The ratings are generally biased towards neutral or positive reviews (3-5), which is possibly indicative of users tendencies to watch movies they know they will like. Interestingly, the rating distribution of most popular movies varies only marginally from the broader dataset; they tend to be higher rated, but only slightly. As expected, the ten highest rated movies have a very different distribution, biased heavily towards perfect reviews.

In addition, looking at Figs. 14, 16, and 15, we notice that the distributions are fairly similar. The only interesting difference comes from rating 3 and 4 for Comedies compared to SciFi and Romance (checking ECDF plot, we see that the difference in 3 and 4 rating for comedies is slightly smaller than the difference in rating for the other two) In fact, even overall the grades are higher for comedies. Probably this is because people are in good mood after watching Comedies and they prefer not to give such low ratings :).

Also, looking at Fig. 5, we notice that Dramas are the most rated movies. This is probably because after a Drama, people reflect upon their own life and realize that they should give back. So they give a rating to Netflix. Or maybe because most of the movies have "drama" in their genre, together with other genre like crime, action etc.

Not that for the violin plots, we chose them to have better density and distribution visualization while not having too many things on top of each other. As the ratings are so discrete as integers, the typically superior stripbox plots are not giving as much aesthetic and clearer info as the violin plots as too many of the dots have the same y axis value and adjusting the alphas for the tranparency does not work as well as one would hope. However, the kernel density from the violin plots can be confusing and is not accurate in that we only have ratings from 1 to 5. In hindsight, we could enforce the cutoffs, too.

3 Matrix Factorization Visualizations [60 points]

Q: How do each of these methods work? How do they differ?
The three methods used in the report are:

- Method 1: This method works by mapping both users and items to a joint latent factor, such that user-item interactions are modeled as inner products in that space. [1]

  Implementation: We have used our own implementation using some help from HW5, as well as the Surprise package with biased = False

$$\underset{U,V}{\operatorname{argmin}} \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right) + \sum_{(i,j)\in S}\left(Y_{i,j} - u_i^T v_j\right)^2$$

- Method 2: Data in general exhibits large systematic tendencies for some users to give higher ratings than others, and for some items to receive higher ratings than others. [1] After all, some products are widely perceived as better (or worse) than others. Therefore, adding bias terms, we can model the global tendency of a movie's average rating and how users rate on average. This is supposed to improve the modelling error.

  Implementation: We have implemented our own algorithm using the previous method as a start.

$$\underset{U,V,a,b}{\operatorname{argmin}} \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right) + \sum_{(i,j)\in S}\left(Y_{i,j} - \left(u_i^T v_j + a_i + b_j\right)\right)^2$$

- Method 3: This method differs from Method 2 in the fact that we add μ, a global bias as averaged over all observed Y. In addition, the biases a and b can be regularized as well.

  Implementation: We have used the Python package "Surprise", as well as our own implementation.

$$\underset{U,V,a,b}{\operatorname{argmin}} \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2 + \|a\|^2 + \|b\|^2\right) + \sum_{(i,j)\in S}\left((Y_{i,j} - \mu) - \left(u_i^T v_j + a_i + b_j\right)\right)^2$$

How did they perform in comparison to one another on the test set?

For each method, we explore using in-house methods and surprise. Within each method, we perform optimization, with or without optimizing the regularization and learning rate. Regarding optimization, we both performed it manually (denoted as * in plot titles) as well as using nevergrad package. Also note that we had more than one set of parameters to generate plots but here we list the once that have the lowest testing errors. Additional plots may be found at the very end of the report.

Custom Implementation	Method 1 (Non-bias)	Method 2 (Not reg bias)	Method 3 (Reg bias)
Training Error	0.444	0.228	0.168
Testing Error	0.793	0.436	0.505
Error Function used	MSE	MSE	MSE
Ks	20	20	30
Epochs run before stop	6	43	45
eta	0.048	0.013	0.051
reg	0	0.069	0.550
Stopping criteria	If E_in doesn't decrease by some fraction esp = 0.0001 of the initial decrease in E_in, stop early		
Optimization Method	Nevergrad	Nevergrad	Nevergrad and manual

Surprise package	Method 1	Method 2	Method 3
Training Error	0.9113	0.92	0.9178
Testing Error	0.8766	0.90	0.88
Error Function used	MSE	MSE	MSE
n_factors	93	87	90
n_epochs	22	23	21
biased	False	True	True
lr_all	0.005	0.022	0.005
reg_all	0.02	0.027	0.02
reg_bu	0.02	0	0.02
reg_bi	0.02	0	0.02
random_state	42		
Optimization Method	Nevergrad	Nevergrad	Nevergrad

*note that MSE for Methods 1,2,3 as computed by the Surprise package does not have a 0.5 factor, compared to the Methods 1,2,3 from the custom implementation.

We notice that for the custom implementation, as expected, Method 2 and 3 outperform Method 1, because in Method 1, the bias is not captured. Interestingly, in Surprise package, the errors of the three methods don't seem to change that much. The errors obtained seem to make sense because they were compared to the baseline from the Surprise's github account. We have spent a tremendous amount of time on understanding the Surprise package and we could not figure out how to make Method 2 and 3 perform better than Method 1. We have noticed many differences in the documentation of Surprise and the actual code from Github, including the "Issues" content on github. In nevergrad, we tried to optimize the lr_all parameterand the reg_all parameter, however, not using the default values made the error go above 1. Probably nevergrad was getting stuck in a local minima. Also, it is surprising that the testing errors from the surprise package are higher than those for training.

Visualization

For the visualization, we took different approaches that either emphasis the clearness of the chosen movies or the comparison of the chosen movies to all of the movies. For the layout, it contains 1) a scatter plot overlay where the x-axis and y-axis correspond to the largest and the second largest U (movie) or V (user) projection, 2) an ECDF (Empirical Cumulative Distribution Function) and histogram for all movie rating counts (popularity), and 3) an ECDF and histogram for all movie average ratings.

For, the scatter plots, all movies area plotted as dots. Each dot represents one movie, where its size is proportional to the number of total rating and its color corresponds to its average rating (the higher the rating, the more bluish the dots). The user components are the grey dots. There is also the overlay of 1) the most rated/popular, 2) the highest rated, where more than 5 rating entries are considered, 3) 10 chosen comedy, 4) 10 chosen romance, 5) 10 chosen sci-fi, and 6) 10 chosen movies as reference. Note that the legends are clickable and the region of the plot is movable for a more interactive exploration. Please refer to the Colab notebook. When interpreting the plots, we recall from HW5 problem 1D that the decomposition should carry the most information from the first k components. Here, we are focusing on the first two, which supposedly to tell us most about the nature of the data.

Even though the scatter plot is our central focus, with the philosophy of "plotting all your data" (credit to Dr. Justin Bois and BEBi103), we would like to keep in mind the original distribution of all the data. ECDF is better than histogram in that it shows all the data points without the bins but can be confusing for those who are not so familiar with it. From the rating counts, it seems there are a couple of super popular ones. From the average rating ones that match the color bar colors, we can see some average rating clustering, which is probably because many movies had enough ratings to even things out.

We chose the movies based on how well we know the content of them as that we can give better analysis and interpretations as listed as the following:

- **Comedy**: 'Toy Story (1995)', 'Four Weddings and a Funeral (1994)', 'Home Alone (1990)', 'Monty Python and the Holy Grail (1974)', 'Princess Bride, The (1987)', 'Groundhog Day (1993)', 'Men in Black (1997)', 'Space Jam (1996)', 'Home Alone 3 (1997)', 'Big Lebowski, The (1998)'
- **Romance**: 'Star Wars (1977), Forrest Gump (1994)', 'Four Weddings and a Funeral (1994)', 'Gone with the Wind (1939)', 'Dirty Dancing (1987)', 'Top Gun (1986)', 'Empire Strikes Back, The (1980)', 'Room with a View, A (1986)', 'Titanic (1997)', 'Jungle Book, The (1994)
- **Sci-fi**: 'Blade Runner (1982)', 'Terminator 2: Judgment Day (1991)', 'Mystery Science Theater 3000: The Movie (1996)', 'Aliens (1986)', 'Terminator, The (1984)', 'Star Trek: The Wrath of Khan (1982)', 'Star Trek III: The Search for Spock (1984)', 'Alien: Resurrection (1997)', 'Day the Earth Stood Still, The (1951)', 'Star Trek: The Motion Picture (1979)'

Method 1

Method 1 overlay with in-house methods

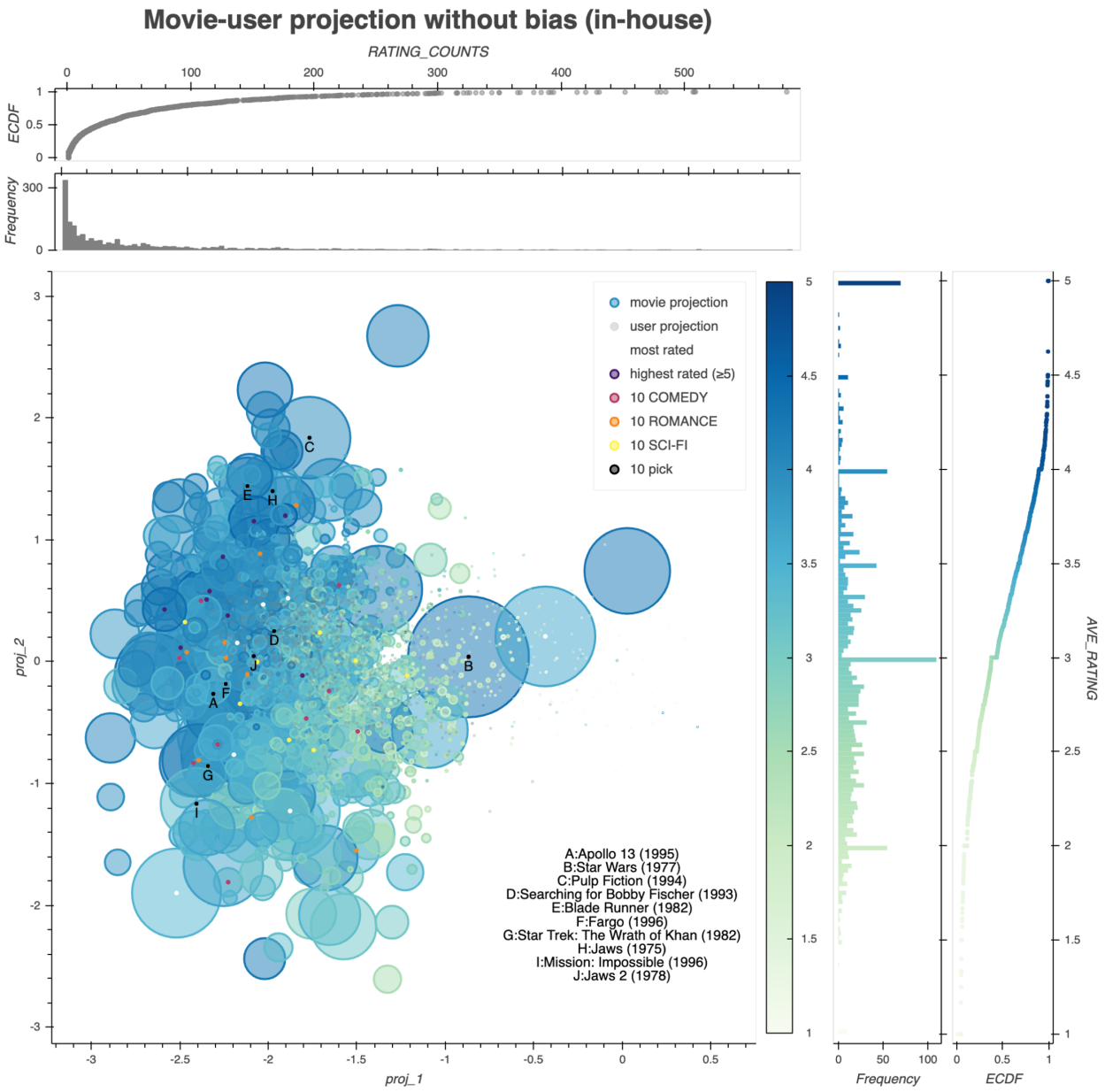

Figure 17: Method 1 overlay, in-house methods, nevergrad optimization, Ks = 20, eta = 0.049, n_epochs = 6, train err = 0.444, test err = 0.793. The color or the size are sort of all blended together and there was not a clear trend. The highest rated ones are sort of clustered around (-2,0) but not quite.

Method 1 overlay with surprise

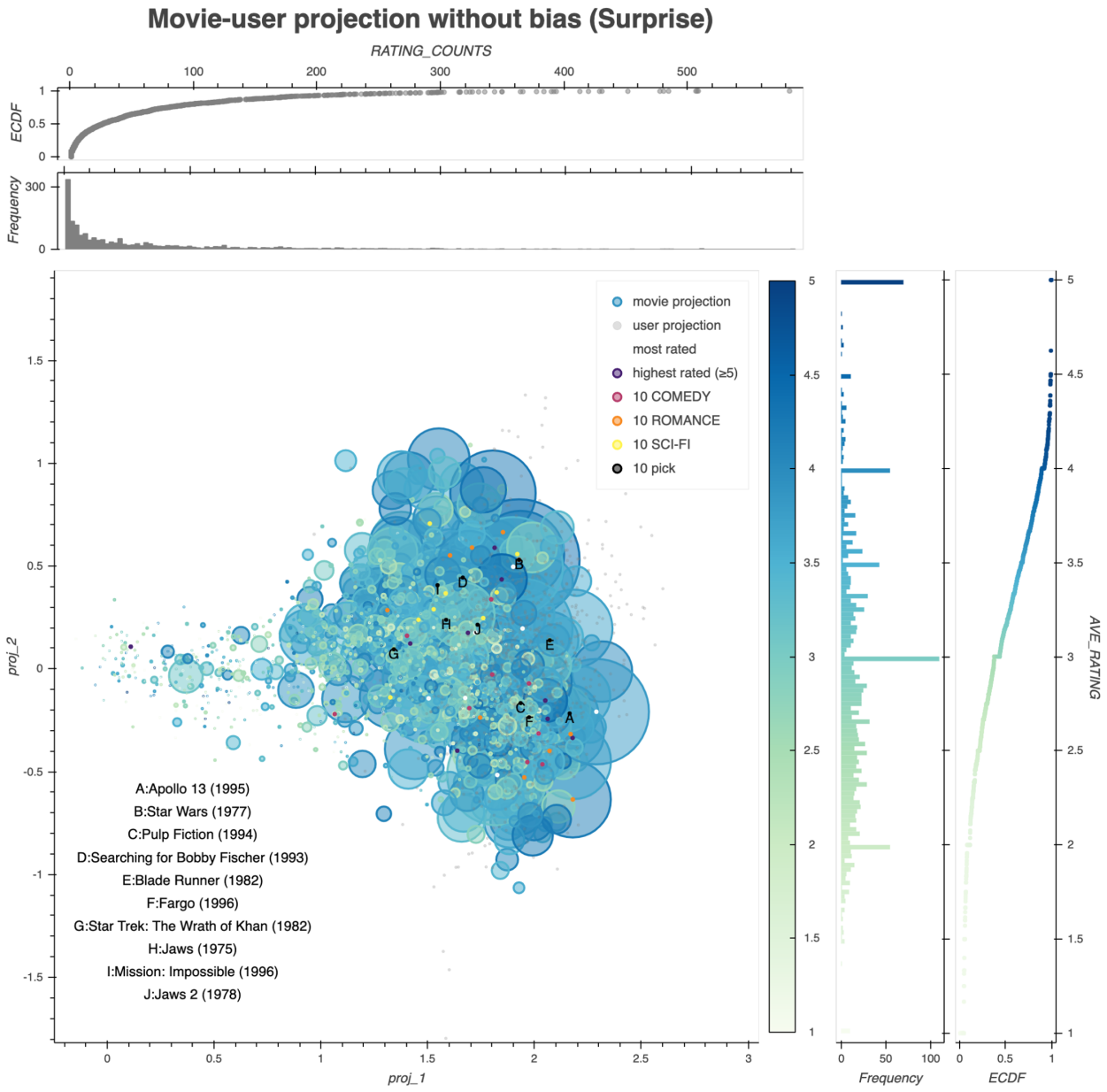

Figure 18: Method 1 overlay, surprise, nevergrad optimization, n_factors = 93, n_epochs = 22, lr_all = 0.005, reg_all = 0.02, train err = 0.912, test err = 0.877, biased=False, turn of bubi. Here we can see some more trend such that the larger and bluer spots (more viewed and higher rated movies) tend to be clustered on the right but still a scramble with other things on top of each other.

Method 2

(a) Any ten movies of your choice from the MovieLens dataset.

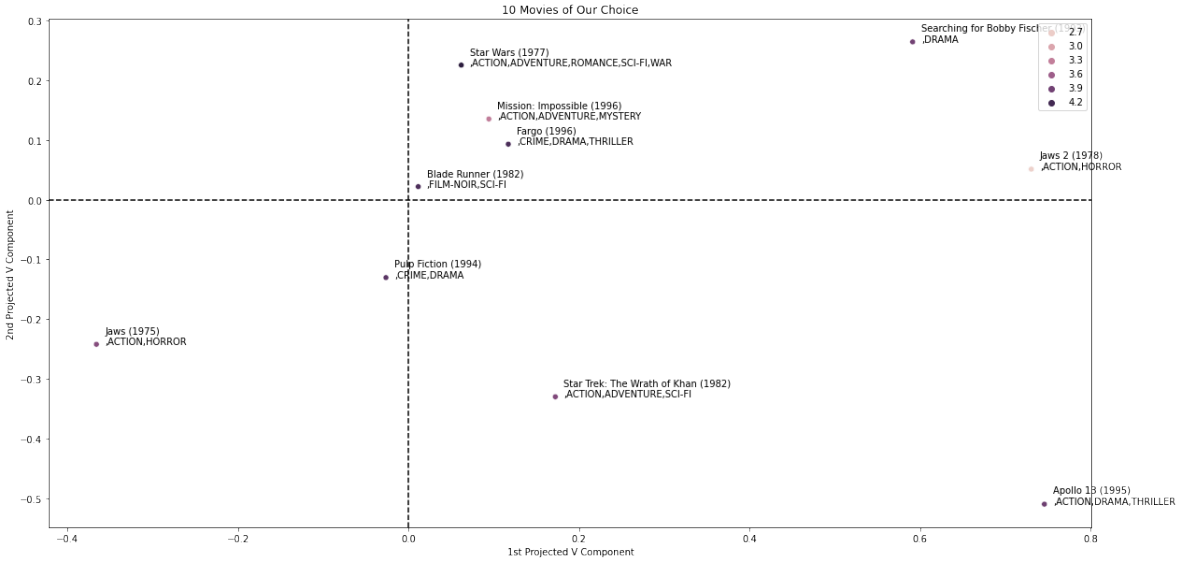

Figure 19: 10 Choice Movies. Although the 'Crime'/'Drama' movies are clusteered as expected, the distance between Jaws and Jaws 2, as well as the lack of clustering of "Star Wars","Star Trek", "Apollo 13" is unexpected.

(b) The ten most popular movies (movies which have received the most ratings).

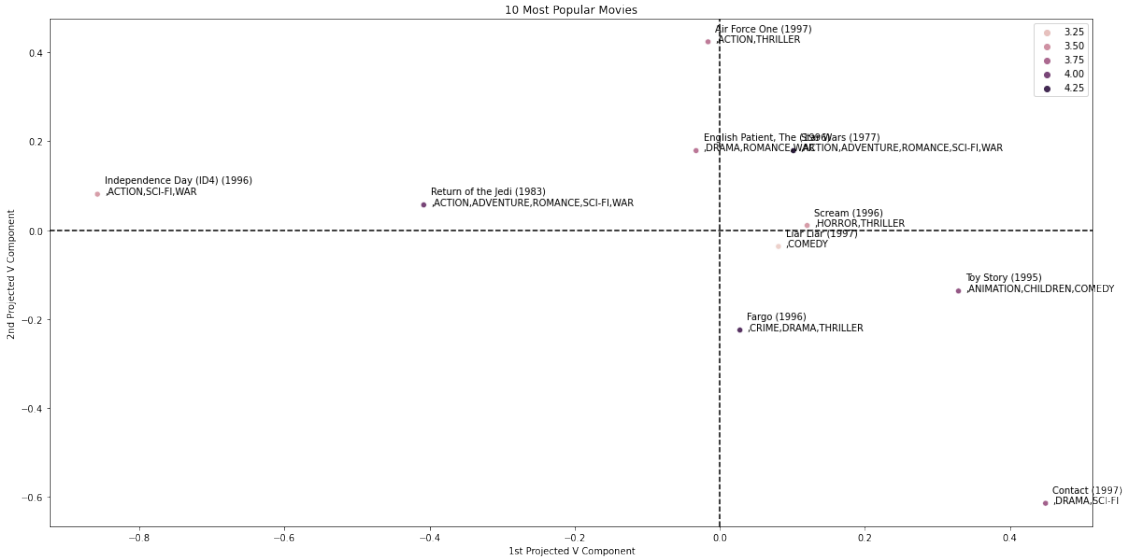

Figure 20: 10 Popular Movies. The LHP sems to favor more action movies, whereas the RHP seems to include more Drama movies.

(c) The ten best movies (movies with the highest average ratings).

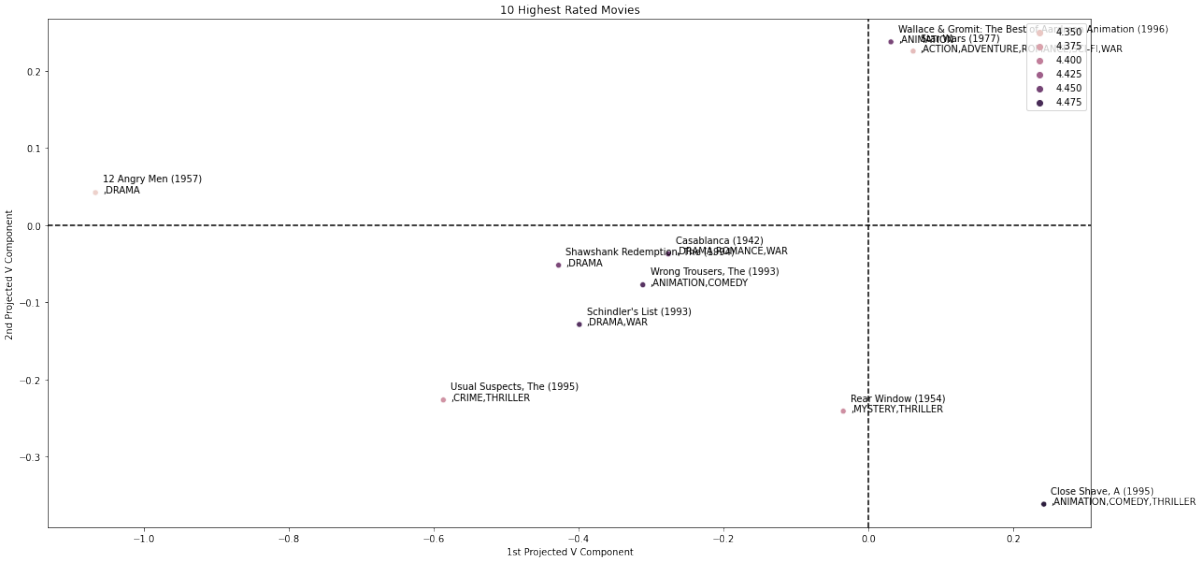

Figure 21: 10 Highest Rated Movies. With the exception of ”The Wrong Trousers”, The War/Drama movies are clustered near the center, with animation movies in the RHP, and 12 Angry Men widely separated.

(d) Genres of Choice

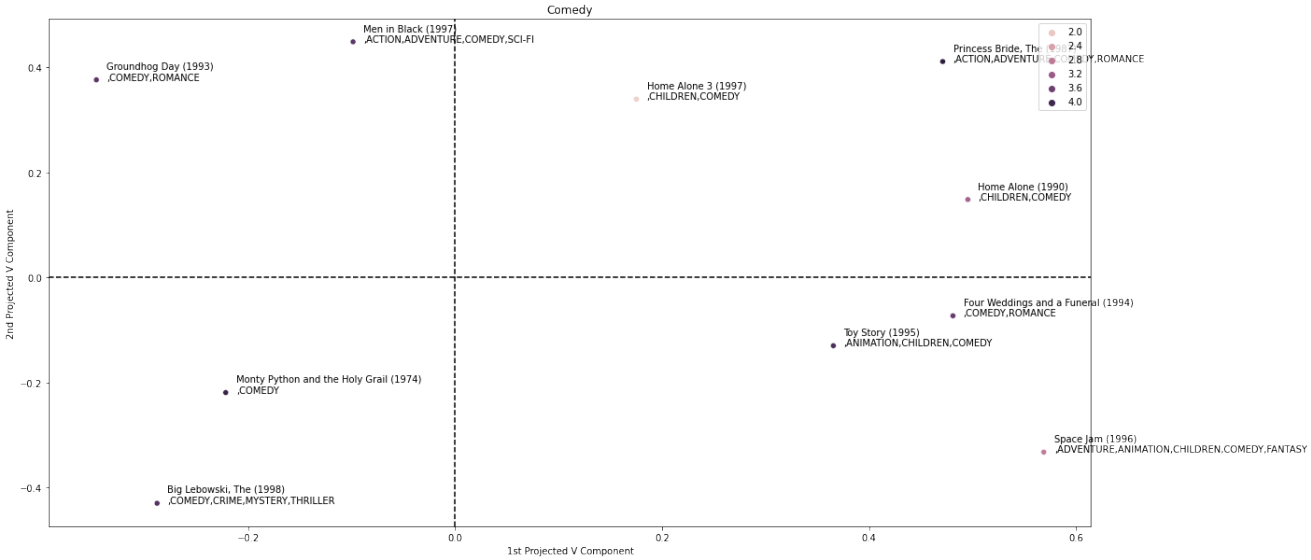

Figure 22: 10 Comedy Movies. The RHP seems to favor children's movies over the LHP. "Home Alone" and "Home Alone 3" appear in the same quadrant.

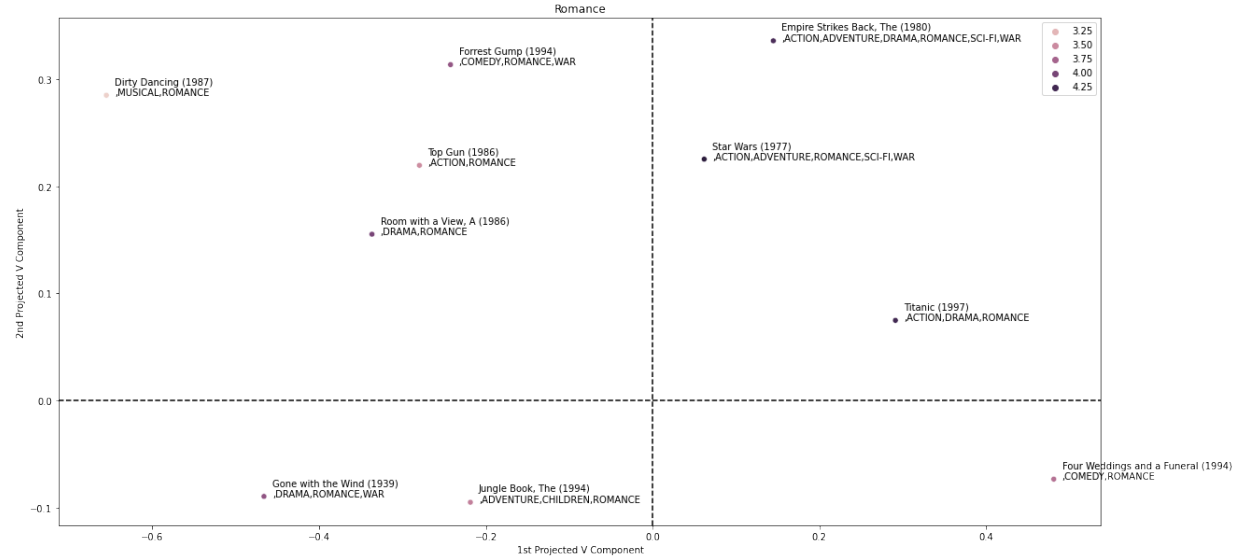

Figure 23: 10 Romance Movies. As expected, "Star Wars" and "The Empire Strikes Back" Are in the same quadrant, along with "Titanic" (another 'Action' movie).

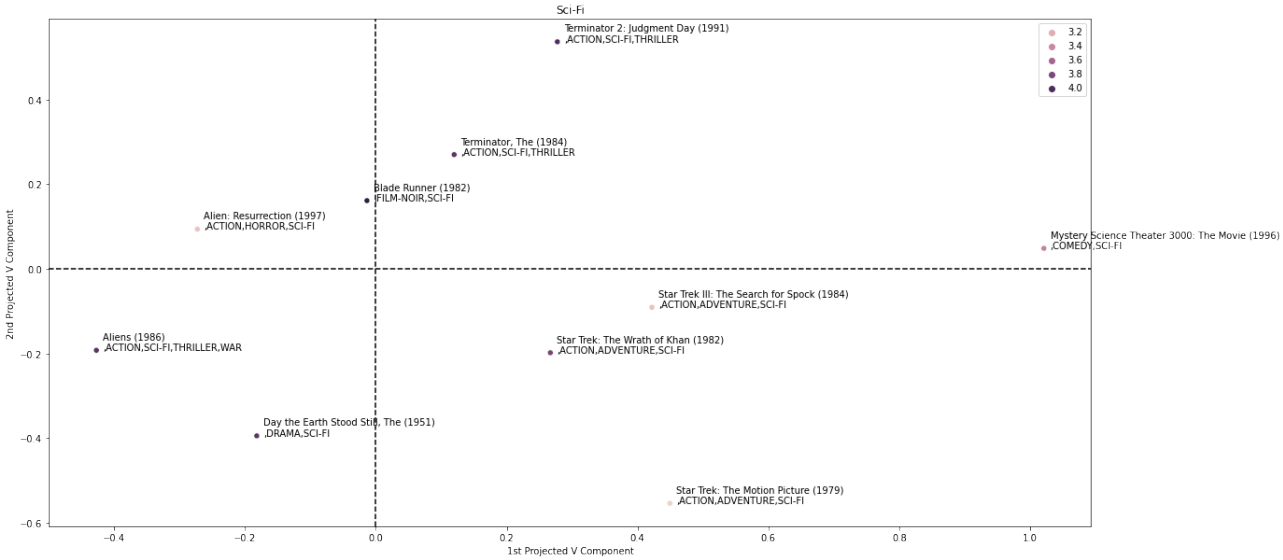

Figure 24: 10 Sci-Fi Movies. Here, we can see ”Terminator” and ”Terminator 2” close to each other, and similarly with ”Alien” and ”Aliens”, and the ”Star Trek” Franchise. ”MST3000”, the least similar to the other movies, is apart from the others.

Method 2 overlay with in-house methods

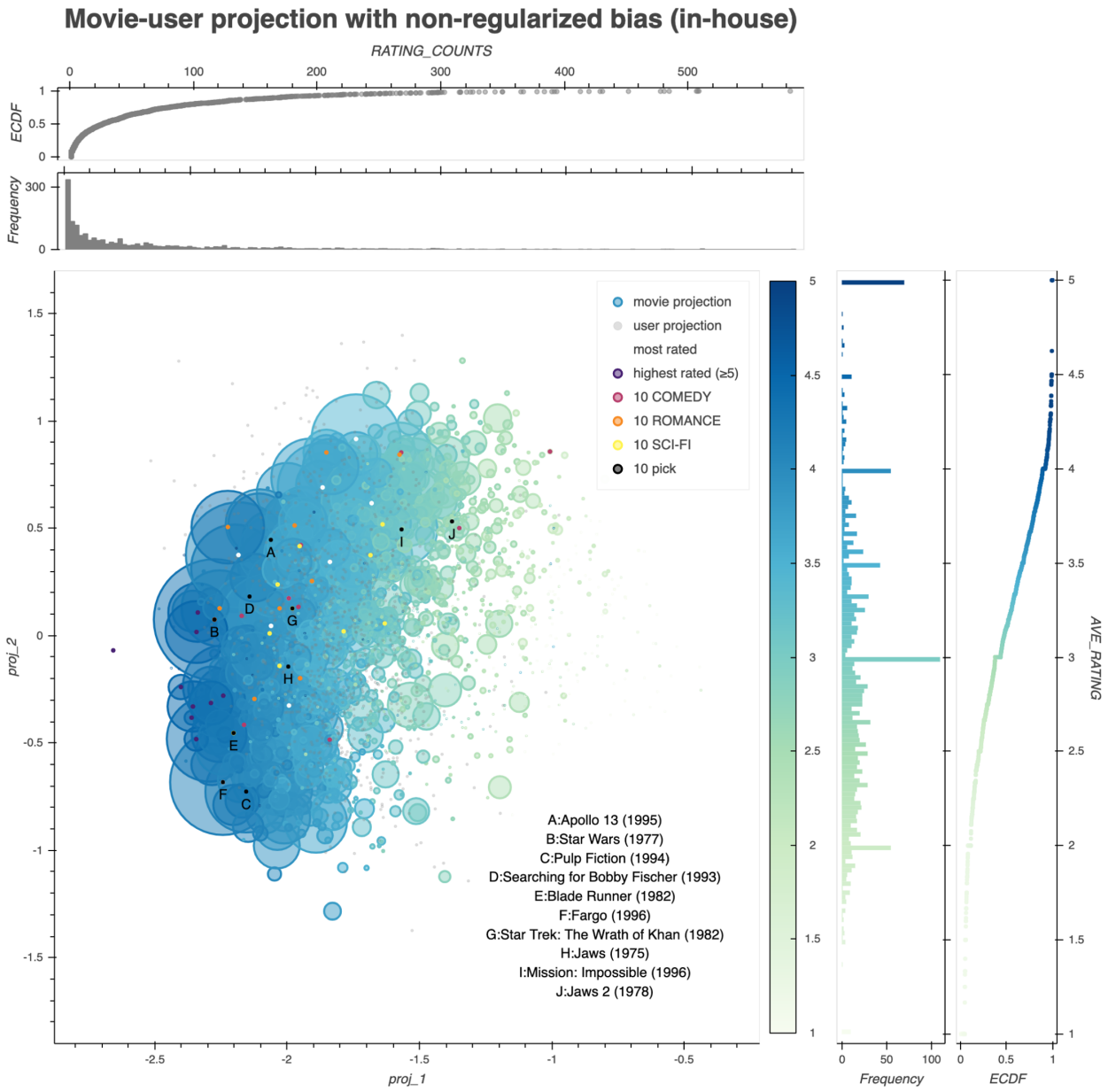

Figure 25: Method 2 overlay, in-house methods, the biases are added as columns of 1s for U and V, never-grad optimization, Ks = 20, eta = 0.013, reg = 0.069, n_epoch = 43, train err = 0.228, test err = 0.436

We can see that the bigger (more popular movies) and bluer(higher rated movies) dots tend to cluster together around the left side while the grey dots (users) scatter more evenly over the place. Noticeably, we see that the top-rated (with at least 5 rating, which is a reasonable cutoff while keeping 80% of the total data) are clustered around (-2.5, 0) in purple dots. There are the most rated ones in white dots are around the line of proj_1 = -2.

Although the latent factor projection appears dominated by the effect of rating count, the labeled movies may indicate additional clustering. While the majority of selected movies are 'Action' or 'Sci-Fi' movies, the two crime/thriller movies ("Fargo", "Pulp Fiction") appear near each other. Additionally, the movies related to space ("Wrath of Khan", "Star Wars", "Apollo 13") are clustered together, which could represent a latent categorization. However, "Searching for Bobby Fischer" appears within this cluster, and is definitely not a space-related movie. Finally, while we expect movies in the same franchise (i.e. "Jaws", "Jaws 2") to be close together, they are widely separated in the 2D projection - it is possible that the significant decrease in quality (and rating) dominates the latent factor representation.

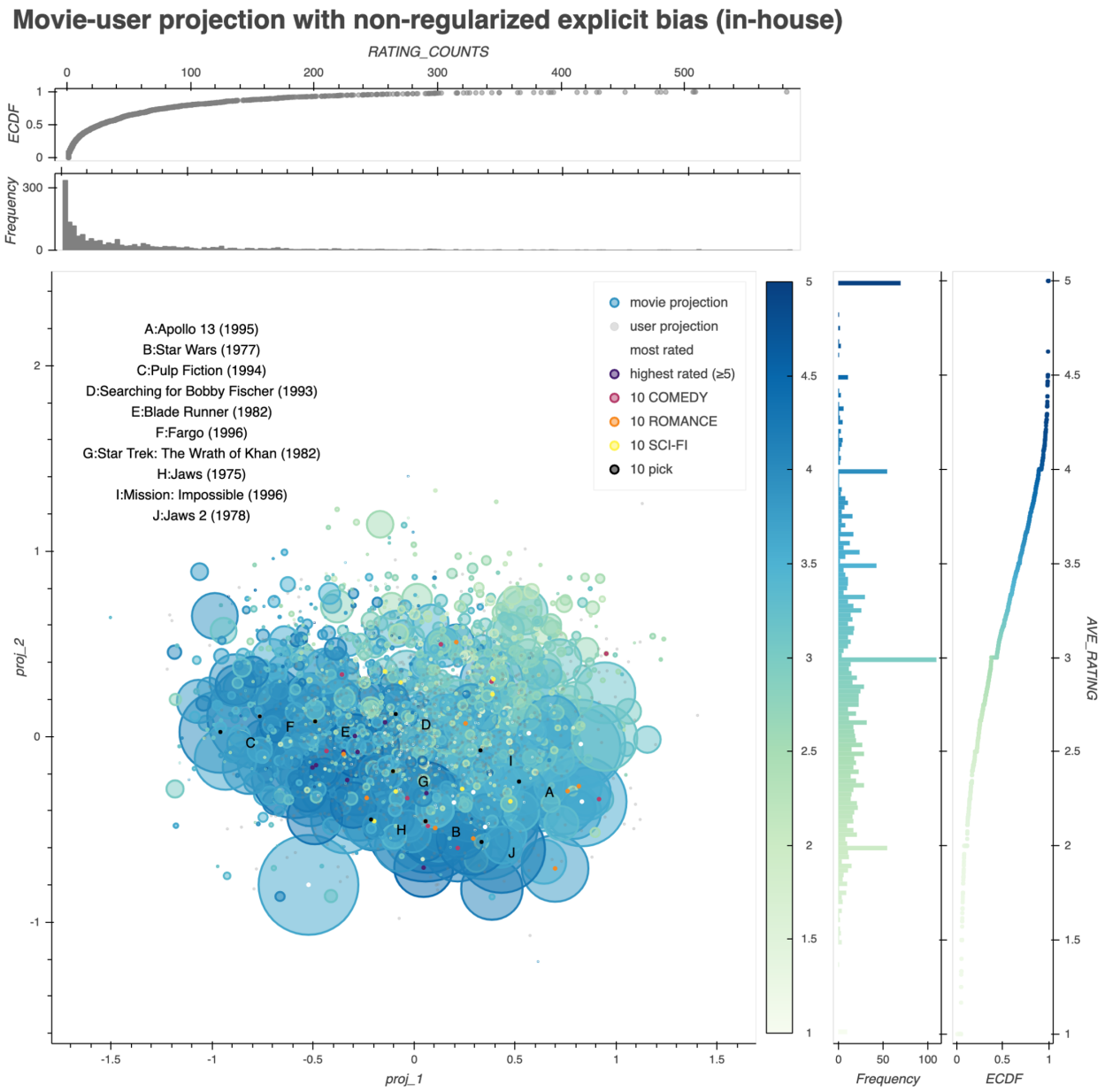

Figure 26: Method 2 overlay, in-house, explicit coded biases, nevergrad optimization, Ks = 20, eta = 0.062, reg = 0.069, n_epoch = 34, train err, 0.205, test err = 0.442

Here we used another way of calculating the components, it is quite similar in trend of clustering but almost feel like a flip about the diagonal line with a slope of 1.

Method 2 with surprise

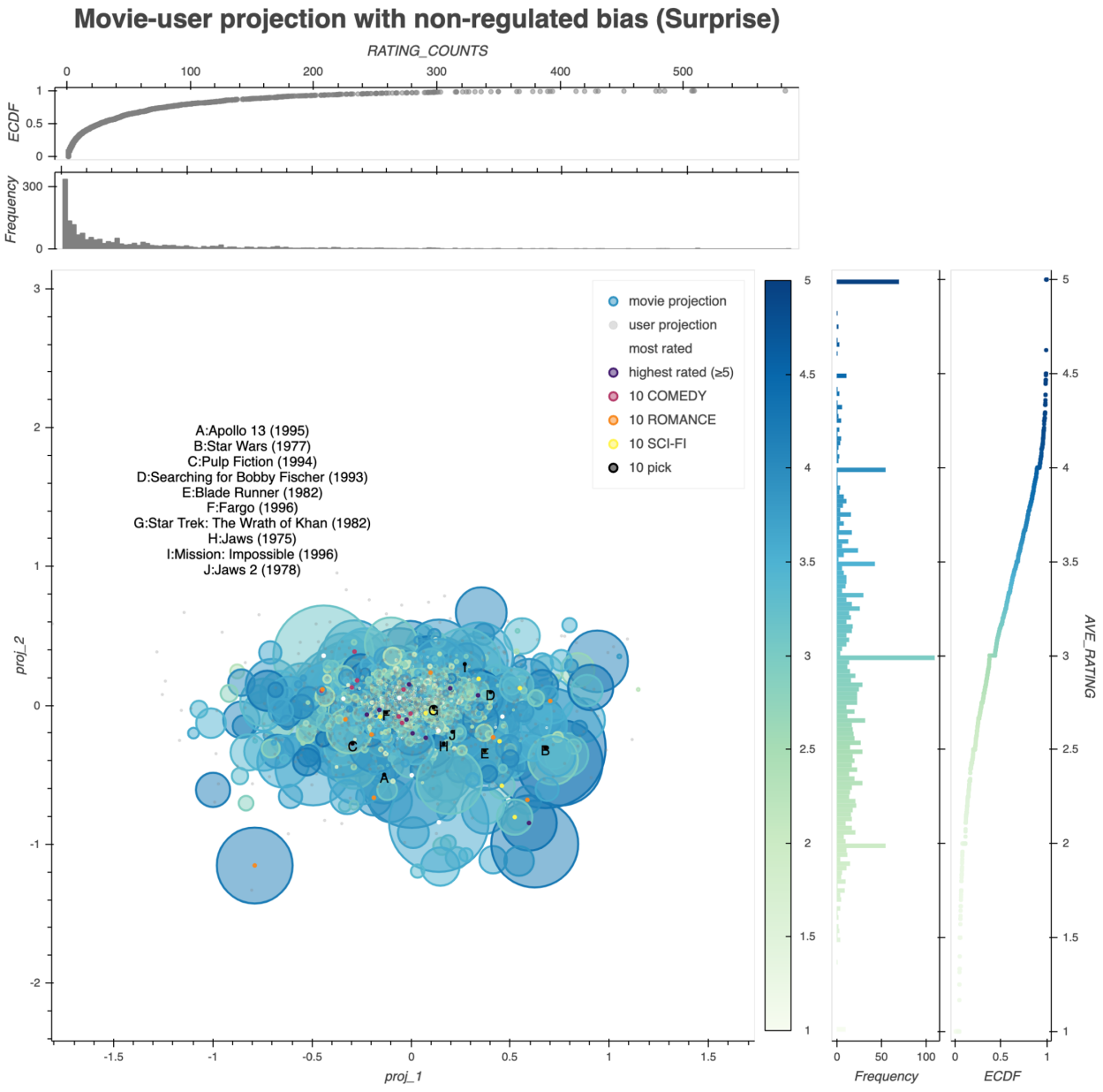

Figure 27: Method 2 overlay, surprise, nevergrad, default learning rate and reg, n_factors = 87, n_epochs = 23, lr_all = 0.005, reg_all = 0.02, biased = True, turned off bubi, train err = 0.922, test err = 0.909

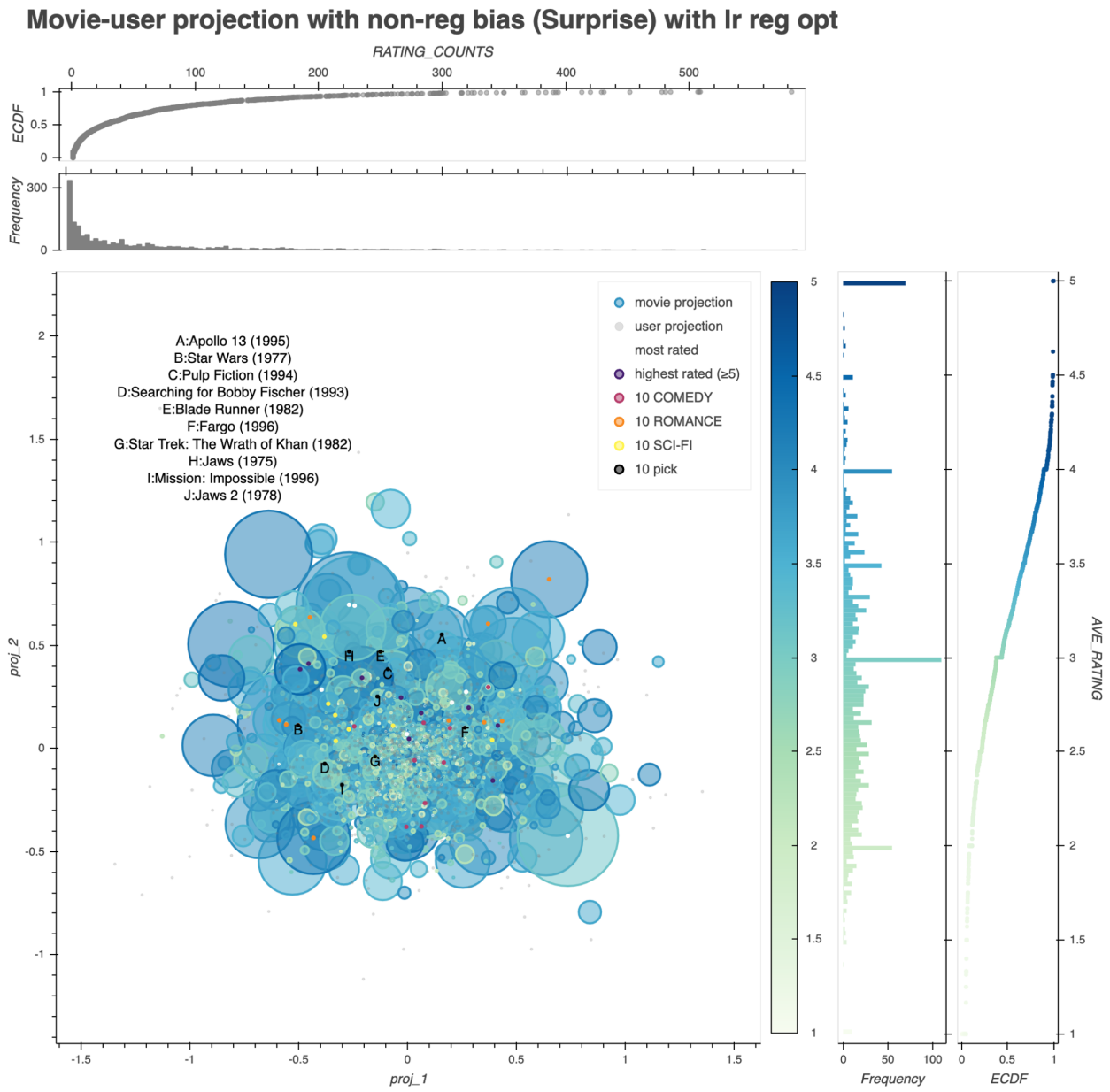

Figure 28: Method 2 overlay, surprise, nevergrad, n_factors = 146, n_epochs = 30, lr_all = 0.022, reg_all = 0.027, biased = True, turned off bubi, train err = 1.116, test err = 1.125

Method 3

(b) The ten most popular movies (movies which have received the most ratings).

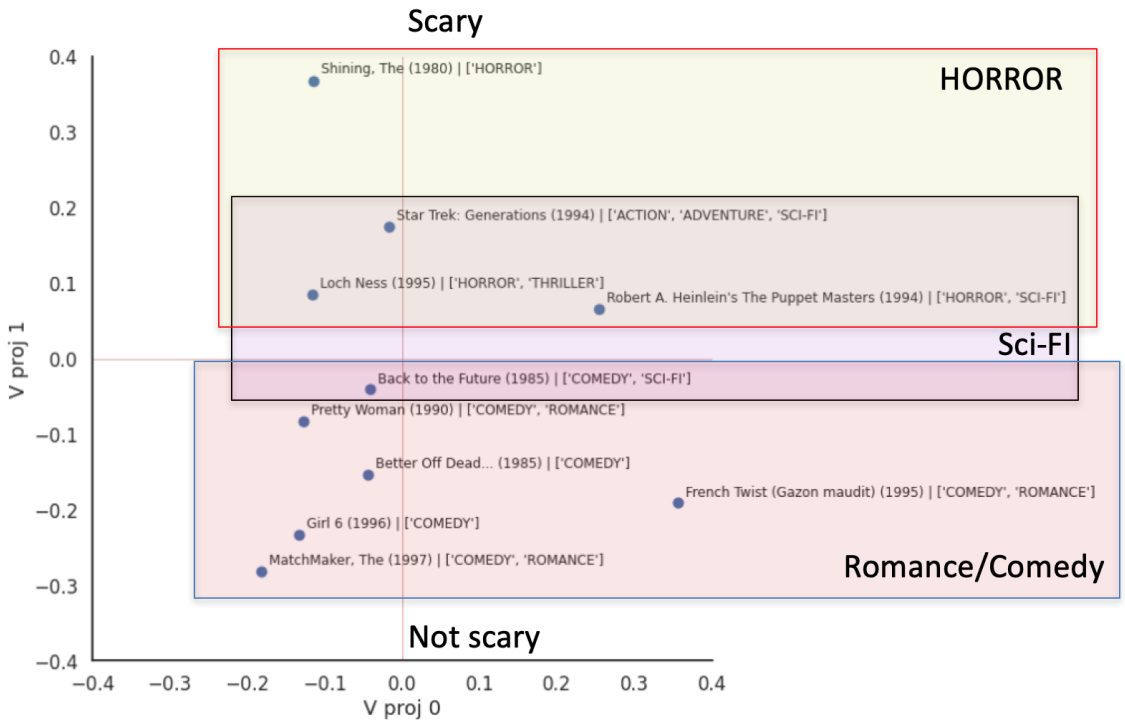

Figure 29: 10 movies of choice with Method 3 - a plot that is easier to interpret

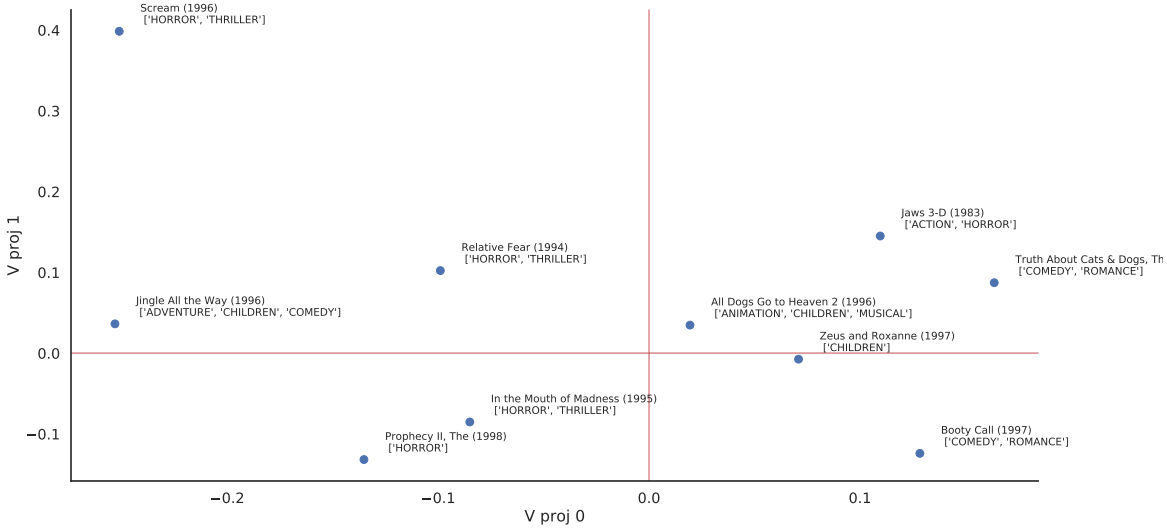

Figure 30: 10 movies of choice with Method 3 - a plot that is harder to interpret

(b) The ten most popular movies (movies which have received the most ratings).

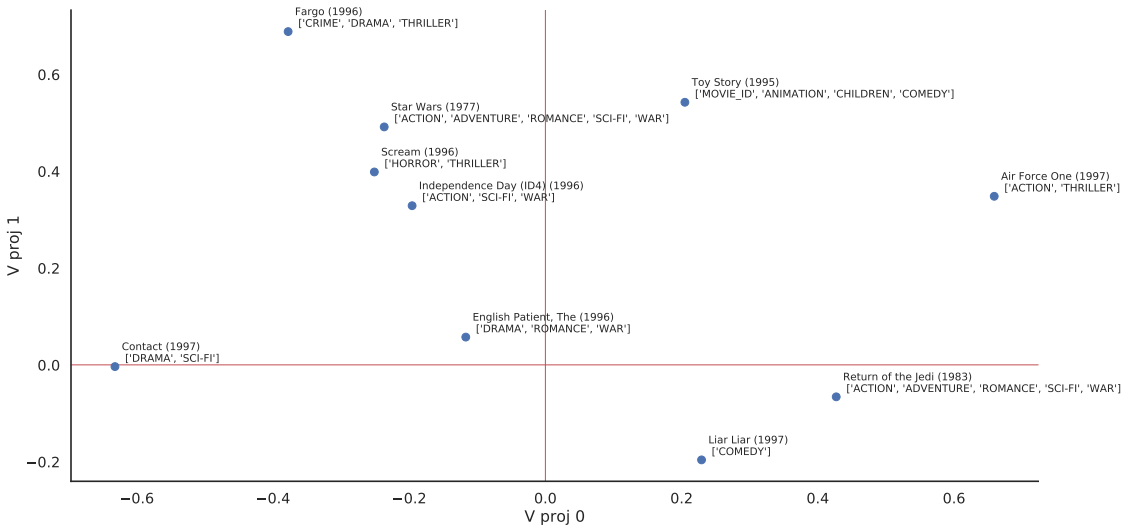

Figure 31: The ten most popular movies

(c) The ten best movies (movies with the highest average ratings).

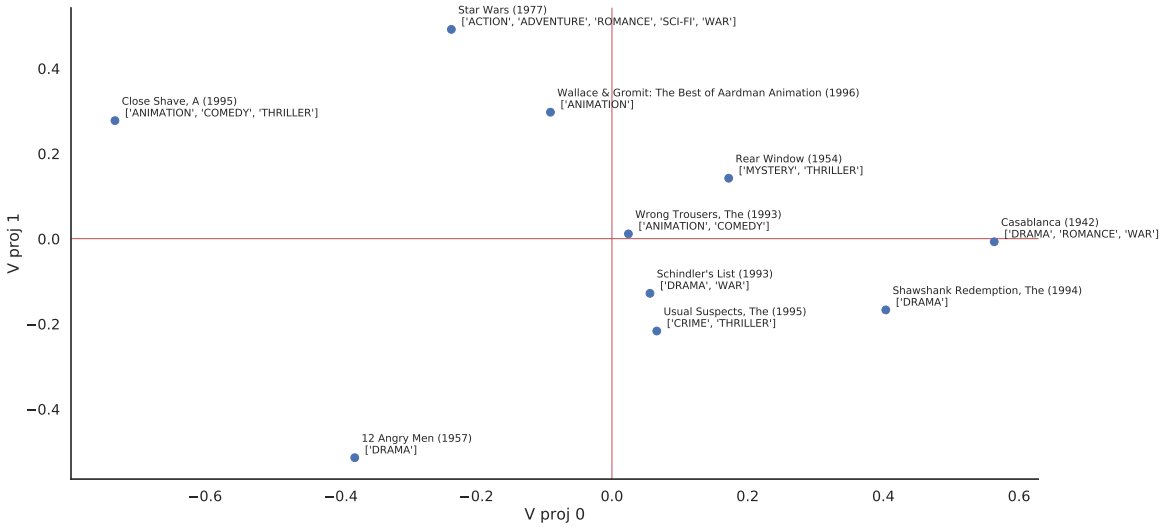

Figure 32: The ten best movies

Method 3 with in-house methods

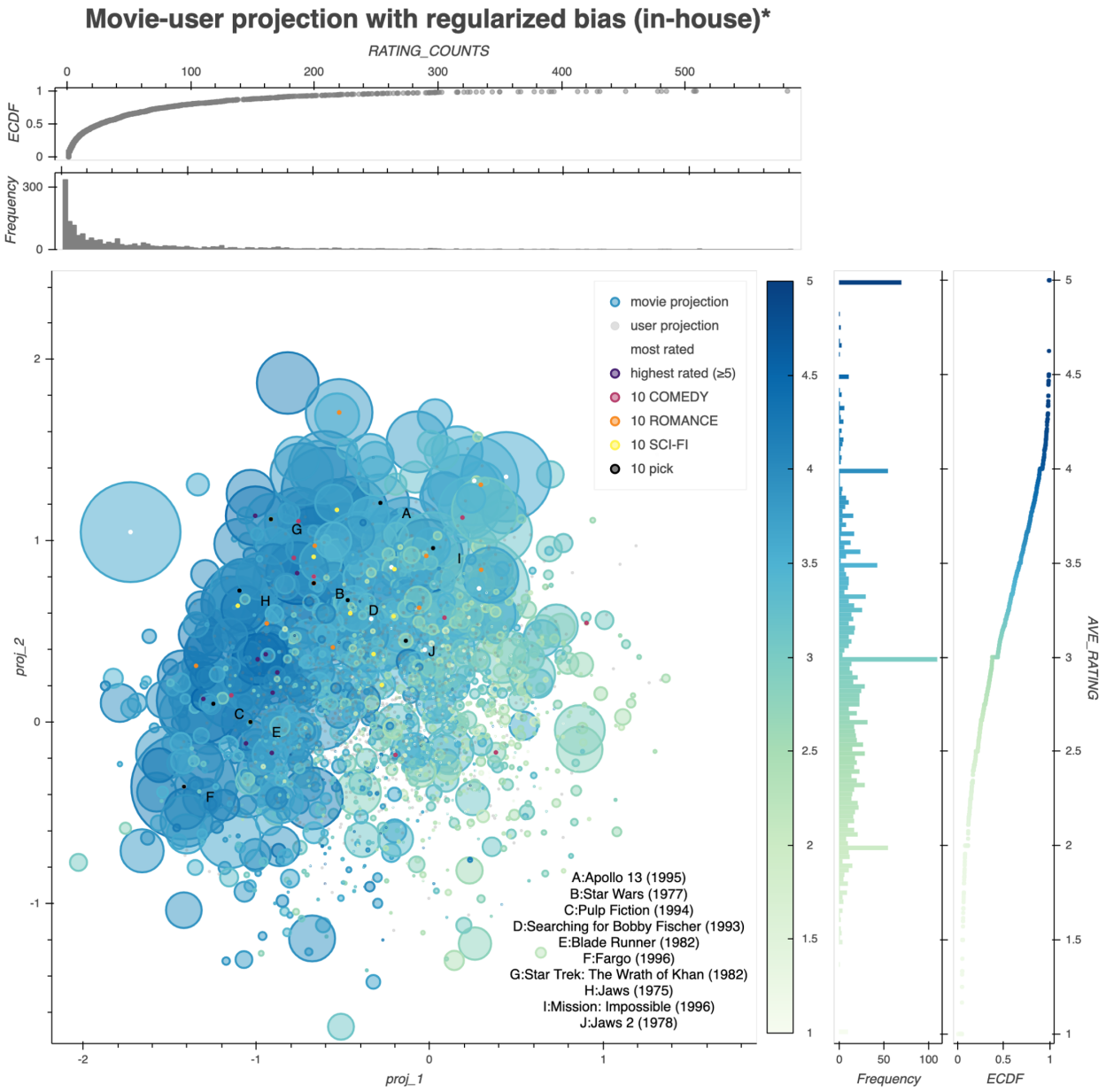

Figure 33: Method 3 overlay, in-house, manual optimization, Ks = 20, eta = 0.03, reg = 0.01, train err = 0.160, test err = 0.603. A little better spread with some size and rating trend

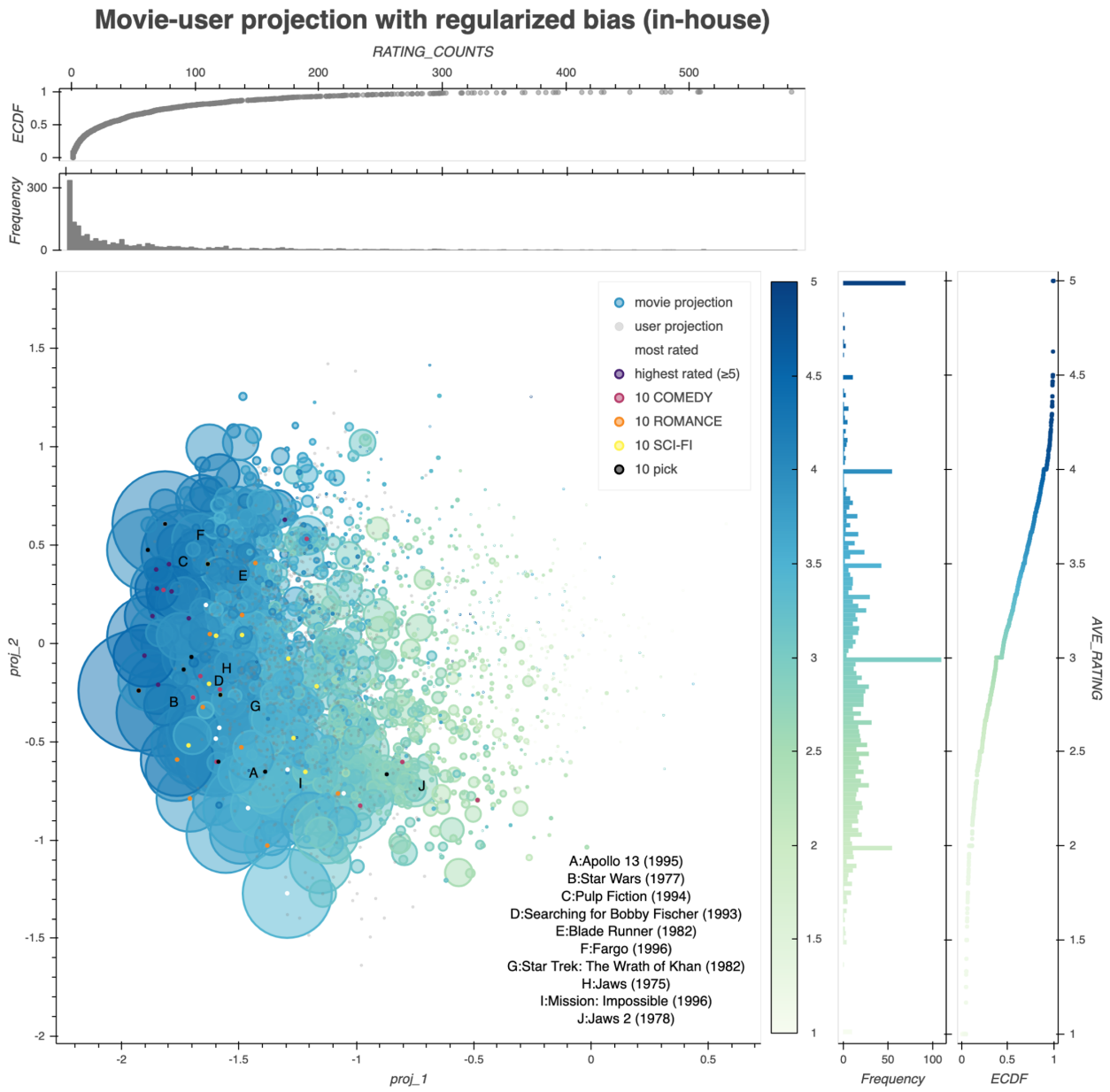

Figure 34: Method 3 overlay, in-house, nevergrad optimization, Ks = 30, eta = 0.051, reg = 0.055, train err = 0.165, test err = 0.508. Similar to previously but with better spread and again the highest rate and the most rated each has a zone with in the blue hear region. There seems to have some romance and sci-fi zones but are not with clean cuts

Method 3 in-house

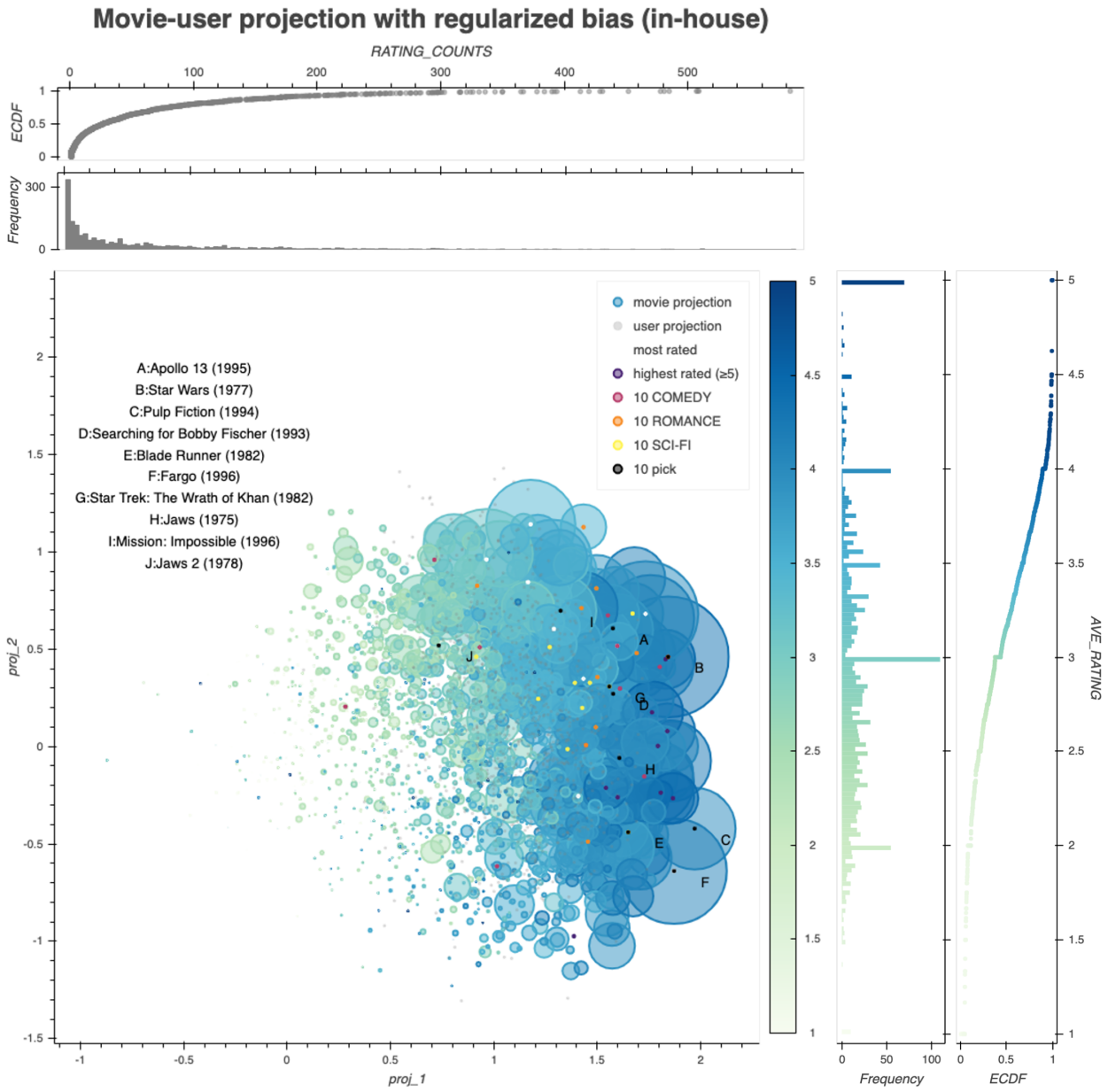

Figure 35: Method 3 overlay with nevergrad optimization

Method 3 surprise

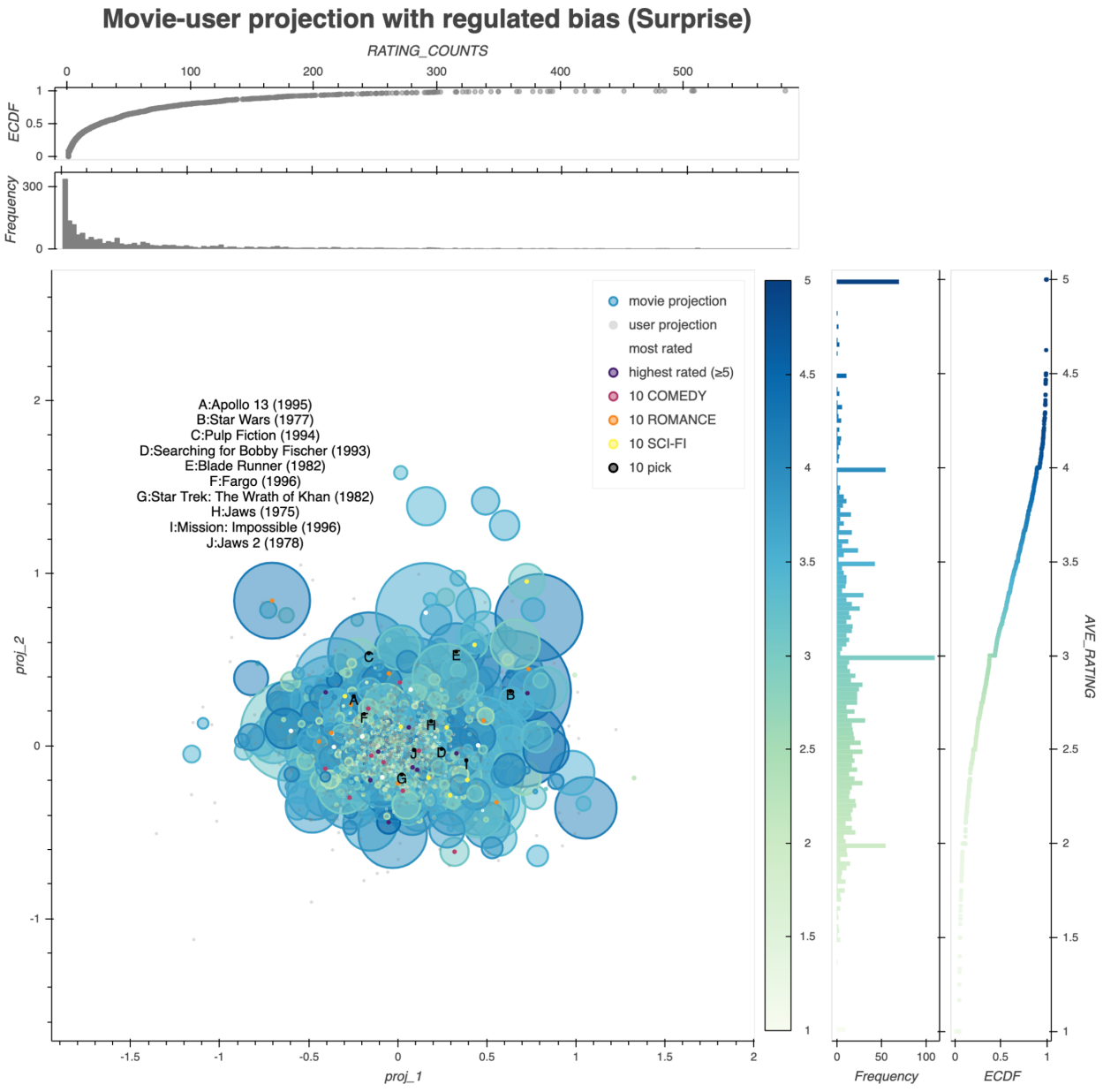

Figure 36: Method 3 overlay, surprise, nevergrad optimization, default learning rate and reg, n_factors = 90, n_epochs = 21, lr_all = 0.005, reg_all = 0.02, train err = 0.914, test err = 0.8855

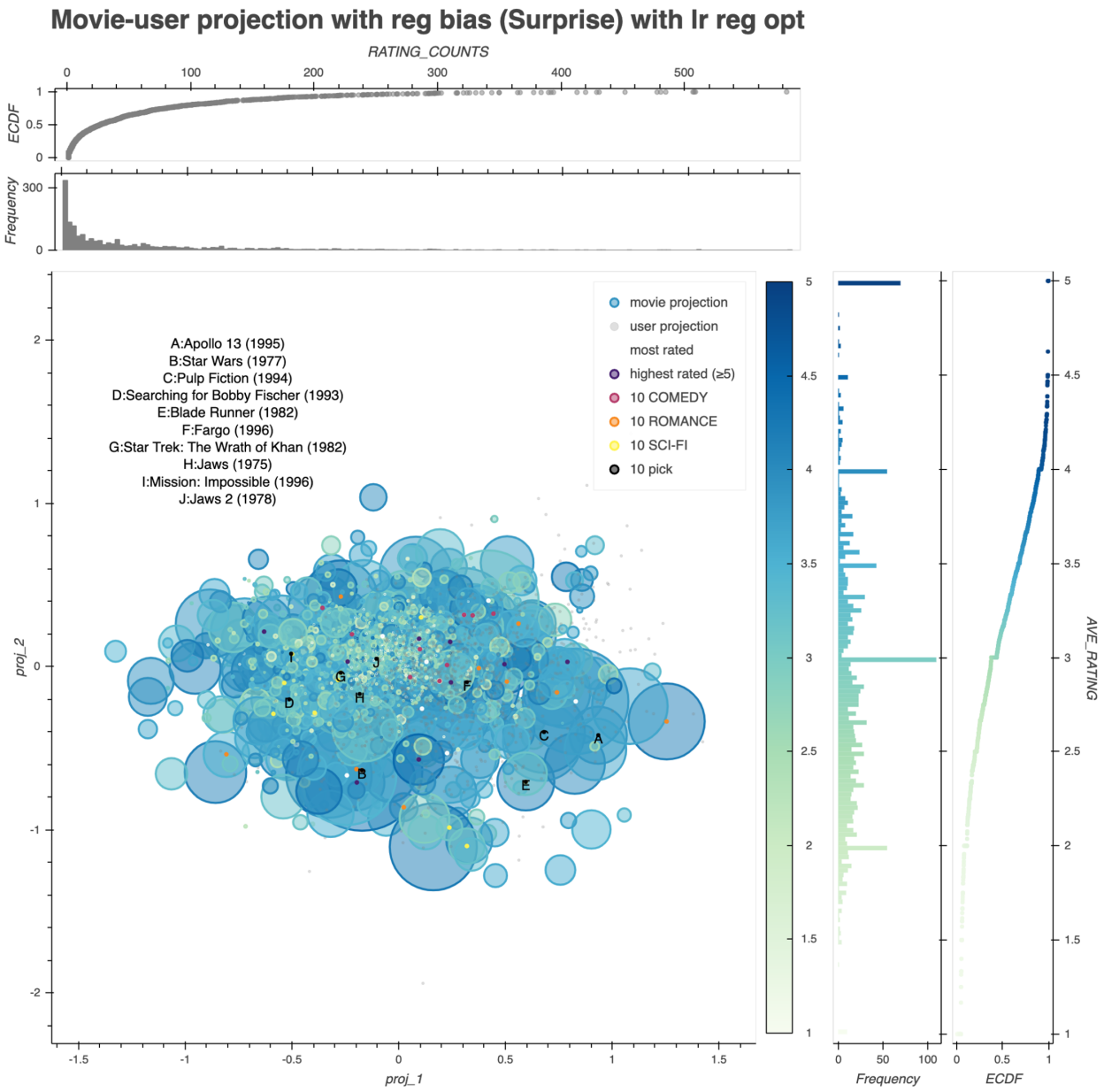

Figure 37: Method 3 overlay, surprise nevergrad, optimized learning rate and reg, n_factors = 147, n_epochs = 21, lr_all = 0.043, reg_all = 0.055, train err = 1.1533, test err = 1.1654

Analysis

Our primary observation is that the resulting 2D projections of the latent factor decomposition is highly sensitive to the choices made during model construction - even slight changes in model formulation resulted in drastically different outcomes. This runs contrary to the expectation that the first two principal components capture the majority of the variability in the data, and ought to be the same regardless of the method used.

Although the rating distribution for the most popular and highest rated movies is quite different, we found no significant difference in their 2D representations across all models, with the notable exception being Method 2 with the non-regularized appended bias, in which the primary component appeared to capture the number of ratings.

While we could identify in certain instances trends within genres (movies in the same franchise tend to be grouped (Star Trek, Star Wars, Alien, Terminator) unless there is a significant difference in quality (Jaws), we were unable to identify trends that reached across genres. In other words, "Romance" movies, "Comedy" movies, and "Sci-Fi" movies, did not appear to cluster in the plane as we would have expected, indicating that neither of the first two principal components capture information about genre. Though arguably there are some slight micro-regions or rather broad overlapping strokes for the genres. We also acknowledge that lots of the movies have complication genre combinations which might not make the cluttering as ideal and clean as one would have imagined.

Comparing the plots with all the movies, with and without biases for both methods, we realized that we can see the trend and the separation of the more popular and better rated movies from those that are rated by very few people and are poorly rated in both the without biases or with unregularized biases cases.

4 Colab and Piazza link [15 points]

The matrix factorization was performed with both the in-house methods as well as with the surprise package. For each, we performed, not biases, not-regulated bias, and regulated bias. Here, we are presenting a layout contains 1) a scatter plot overlay where the x-axis and y-axis correspond to the largest and the second largest U (movie) or V (user) projection, 2) an ECDF (Empirical Cumulative Distribution Function) and histogram for all movie rating counts (popularity), and 3) an ECDF and histogram for all movie average ratings.

For, the scatter plots, all movies area plotted as dots. Each dot represents one movie, where its size is proportional to the number of total rating and its color corresponds to its average rating (the higher the rating, the more bluish the dots). The user components are the grey dots. We can see that the bigger (more popular movies) and bluer(higher rated movies) dots tend to cluster together around the left side while the grey dots (users) scatter all over the place. Noticeably, we see that the top rated (with at least 5 rating, which is a reasonable cutoff while keeping 80% of the total data) are clustered around (-2.5, 0) as the purple dots show. This indicates that there are the most rated ones are around the line of proj_1 = -2

Although the latent factor projection appears dominated by the effect of rating count, the labeled movies may indicate additional clustering. While the majority of selected movies are 'Action' or 'Sci-Fi' movies, the two crime/thriller movies ("Fargo", "Pulp Fiction") appear near each other. Additionally, the movies related to space ("Wrath of Khan","Star Wars","Apollo 13") are clustered together, which could represent a latent categorization. However, "Searching for Bobby Fischer" appears within this cluster, and is definitely not a space-related movie. Finally, while we expect movies in the same franchise (i.e. "Jaws", "Jaws 2") to be close together, they are widely separated in the 2D projection - it is possible that the significant decrease in quality (and rating) dominates the latent factor representation.

Even though the scatter plot is the our major focus, with the philosophy of "plotting all your data" (credit to Dr. Justin Bois and BEBi103), we would like to keep in mind the original distribution of all the data. ECDF is better than histogram in that it shows all the data point without the bins but can be confusing for those who are not so familiar with it. From the rating counts, it seems there are a couple super popular ones. From the average rating ones that matching the colorbar colors, we can see that there are some average rating clustering, which is probably due to not that many movies had enough ratings to even things out.

P.S. To get better visualization experience, the overlayed dots can be turned on (solid) and off (shaded) in the Colab notebook.

Also, to support the initiative of our friends, the Super Unsupervised, Panda Team decided to fit a jellyfish to the plot. For this highly uninterpretable plot, what's more fitting than a jellyfish that wonders around lost in the realm of the sea. This also fits really well the types of movies we chose, for example Jaws!

The bubbles fit the scenery, as well :)

The link to Colab notebook, the Piazza post and our repo.

Other plots:

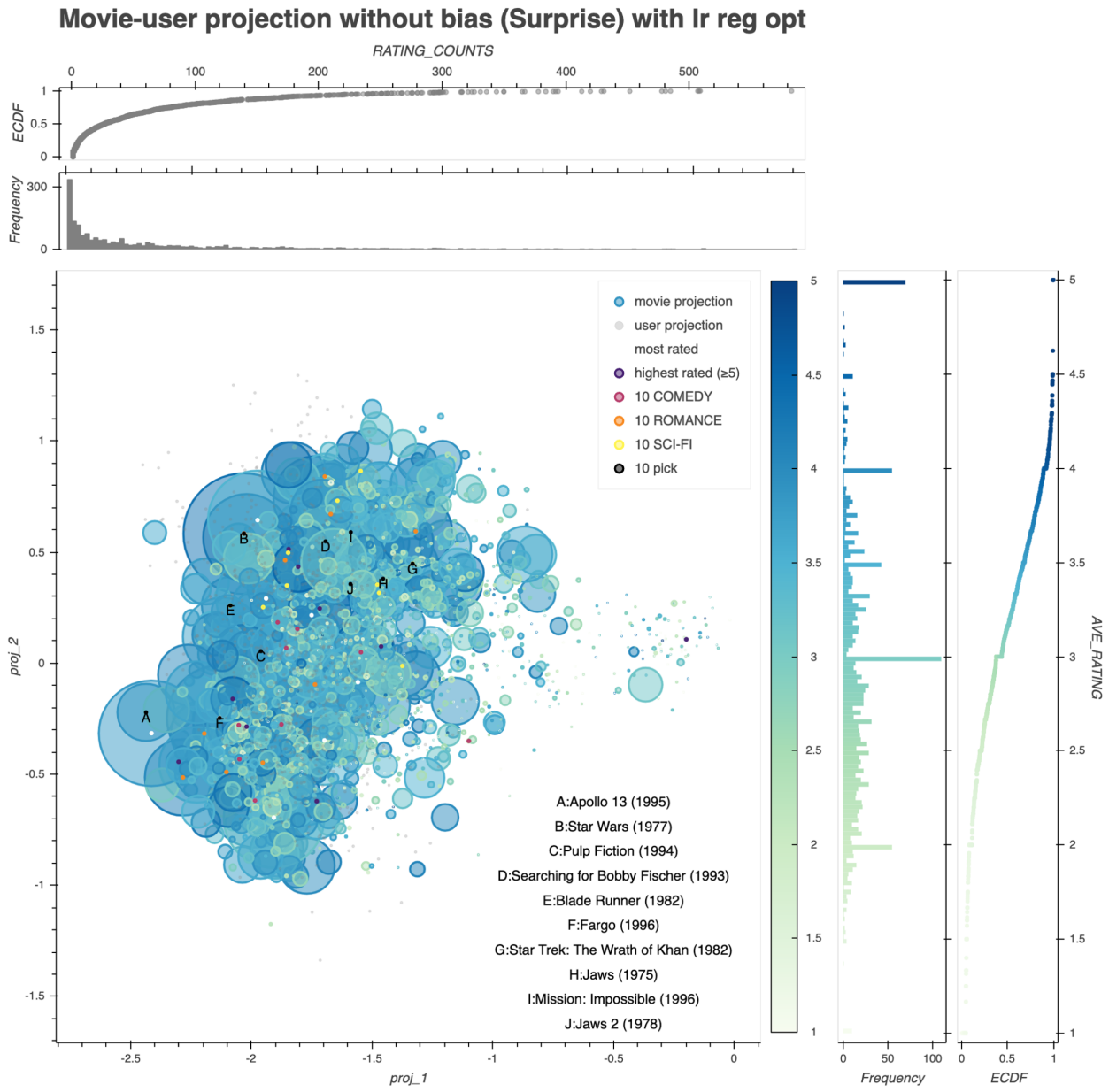

Figure 38: Method 1 overlay, surprise, nevergrad optimization, n_factors = 125, n_epochs = 42, lr_all = 0.051, reg_all = 0.05, train err = 0.945, test err = 0.927, biased=False, turn of bubi

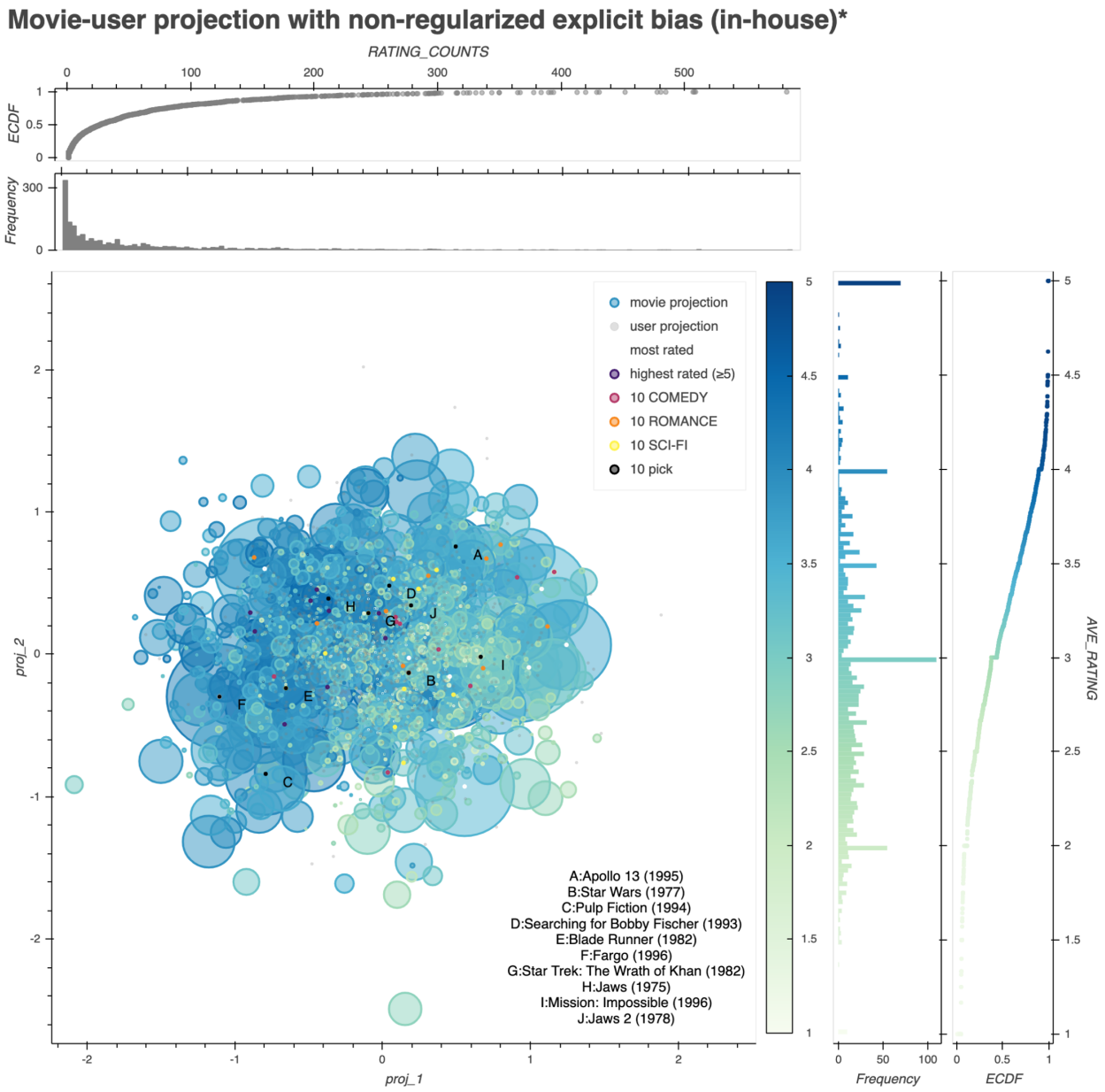

Figure 39: Method 2 overlay, in-house, explicit coded biases, manual optimization, Ks = 20, eta = 0.03, reg = 0.01, n_epoch = 32, train err = 0.157, test err = 0.600