Data Quality Diagnosis

Yitong Liu: yl5549 Yuhao Wang: yw3924 Jiexin Pan jp4479

April 23, 2024

1. Context

In experimental research, the SRM and distribution mismatch analysis method are critical measures used to assess whether there is a statistically significant difference of sample and feature distributions between treatment and control groups in experiments. Ideally, for a balanced experimental design, the composition of the treatment and control groups should be equivalent, typically envisioned as a 50% to 50% ratio. However, deviations from this ideal distribution can occur due to various sampling or allocation errors. [1]

Such discrepancies between the expected and actual proportions of group allocation are significant as they can introduce biases, undermining the integrity of the experiment. These biases can potentially skew the outcomes, leading to misleading conclusions about the effectiveness or impact of the intervention being tested. Therefore, it is imperative to detect any significant deviations in group composition—referred to as SRM or distribution mismatch—prior to conducting the experiment.[2]

The challenge is how to correctly detect SRM and distribution mismatch, which involves statistical tests that quantify the degree to which the groups deviate from the expected. By identifying and adjusting for these discrepancies early, researchers can ensure a more reliable and valid experimental framework, thereby enhancing the accuracy of the study's findings.

2. Executive summary

The complete elimination of SRM and distribution mismatch is unachievable. However, it is possible to detect them through rigorous statistical methodologies. In our current research framework, we are employing parametric methodologies like regression analysis and two sample t-tests, as well as nonparametric methodologies like goodness of fit chi-square tests and K-S tests, to detect potential imbalances in our data.

3. SRM for categorical data

3.1 Use Regression [3] for possible attributes which affect the treatment

First of all, we generalize our own test data set and create three different features: “countries”, “user_status”, and “day_time”. In experiment 1, we do not adjust balance for any group, while in experiment 2, we adjust the balance for either country = “Manhattan”, user_status = “Dasher”, or “day_time” = night. In experiment 3, we only change the balance for “user_status”. After these preparations, we run the generalized linear model from “statsmodels.formula.api” and get the p-values. The null hypothesis is that the true coefficient is zero. If the calculated p-value is small, then we reject null, which indicates that the treatment assignment is not balanced. We found out that if we set the ratios to be large, for example 77% vs 23% like in experiment 2, then the p-values become extremely small. This means that our betas become close enough to zero, i.e. the features have no effect. On the other hand, if we adjust the ratios to be a little bit more balanced, for example, 67% vs 33% as in experiment 3, then we can find out that there is one p-value that is in normal range while the others still remain extremely small. While the balanced group’s p-values are all large enough (from 0.553482 to 0.983849). This indicates that the betas have an effect.

	statistic	pvalue	df_constraint
Intercept	0.351115	0.553482	1
county	2.083597	0.555238	3
user_status	0.032565	0.983849	2
day_time	0.564852	0.753952	2

Figure 1: Result for experiment 1

	statistic	pvalue	df_constraint
Intercept	0.565668	4.519855e-01	1
county	17.851929	4.718854e-04	3
user_status	39.760230	2.323677e-09	2
day_time	21.508340	2.135616e-05	2

Figure 2: Result for experiment 2

	statistic	pvalue	df_constraint
Intercept	20.098404	7.355803e-06	1
county	0.160321	9.837254e-01	3
user_status	53.849654	2.026265e-12	2
day_time	1.080320	5.826550e-01	2

Figure 3: Result for experiment 3

3.2 Use Chi Squared Method [4] for possible attributes which affect the treatment

Under this method, we are using the exact same data that we have generated before. We have the control group and treatment group also the same as before. Then we create lists of unique values for 'country', 'user_status', and 'day_time' and combine them into dictionaries. After the preparation, we perform chi-squared tests by iterating through each key-value pair, and each key in the dictionary and calculating each unique value's frequency in both control and treatment group. By using “scipy.stats.chisquare”, we perform the chi-square test and get the p-value result. Based on research, the current threshold for p-value under SRM is 0.001[5]. The result of our experiment is all smaller than this threshold, which indicates the statistical significance of the association between each categorical variable and some outcome variables, likely represented by 'control' and 'treatment' groups.

	p_value
Queens	1.625152e-03
Bronx	1.448363e-07
Manhattan	8.583768e-18
Brooklyn	2.059063e-06
Dasher	5.500364e-28
Customer	1.036095e-04
Restaurant	7.016286e-05
Afternoon	1.736007e-06
Night	2.157935e-21
Morning	2.461245e-06

Figure 4: Chi-Squared test result for categorical feature

4. Distribution mismatch for continuous features in A/B testing

So far we've investigated the ratio mismatch for categorical features and discussed how to deal with the issue. Now I think it's time to expand our result to continuous features. However, instead of saying "ratio mismatch", we use "distribution mismatch" for this characteristic since it's unable to define a "ratio" for continuous data. Distribution mismatch here, unlike the common one that occurs when the training dataset and test dataset are not drawn from the same distribution, is to detect whether the distribution of a specific feature is different between control group and treatment group.

To illustrate "distribution mismatch" and explore potential methodologies to tackle the issue, we import an A/B testing dataset from Kaggle.[6] This data contains "campaign name", "date", "spend", "number of impressions", "number of website clicks", "number of add to cart", etc. The initial target of this dataset is to compare two campaigns — control and treatment — against each other to determine which one helps the company to get more customers, but our goal is to explore whether there is a distribution mismatch for the same feature between control and treatment groups.

4.1 Regression Analysis

Same as before, we implement Regression analysis first, here is how we define our regression checker for distribution mismatch:

```python
# treatment parameter is the column name which indicates if it is treatment or not (string)
# features parameter should be a (list)
def DM_regression_checker(df, features):
    features_formula = ' + '.join([f'Q("{feature}")' for feature in features])
    formula = f'treatment ~ {features_formula}'
    # fit the regression
    model = smf.glm(formula, data=df).fit()
    # get the p-values for the main effect using a Wald test
    wald_p_values = model.wald_test_terms(scalar=True).table
    return wald_p_values
```

✓ 0.0s

Figure 5: function code for regression checker

As we can see, we construct a Generalized Linear Model (GLM) and fit it with "treatment" (which contains "0" and "1" corresponding to either control or treatment group) as the dependent variable and other features as independent variables. Then we use a Wald test to test whether the coefficients for the feature are significantly different from zero in order to determine whether there is a distribution mismatch between control and treatment group for that feature. The following is the result for our regression analysis:

	P-Value for regression analysis
Spend [USD]	2.978080e-03
# of Impressions	1.036683e-06
Reach	1.155623e-07
# of Website Clicks	1.148164e-01
# of Searches	2.553987e-01
# of View Content	6.339643e-01
# of Add to Cart	2.146654e-05

Figure 6: Result of regression analysis for continuous features

Notice that we choose 0.001 as the threshold of p-value. In other words, A significant distribution mismatch will be reported if we observe a p-value less than 0.001.

As we seen from the analysis, “# of Impressions”, “Reach”, and “# of Add to Cart” show a significant distribution mismatch between control group and treatment group. To visualize the result, we also created histograms for distributions of these features in both control groups and treatment groups.

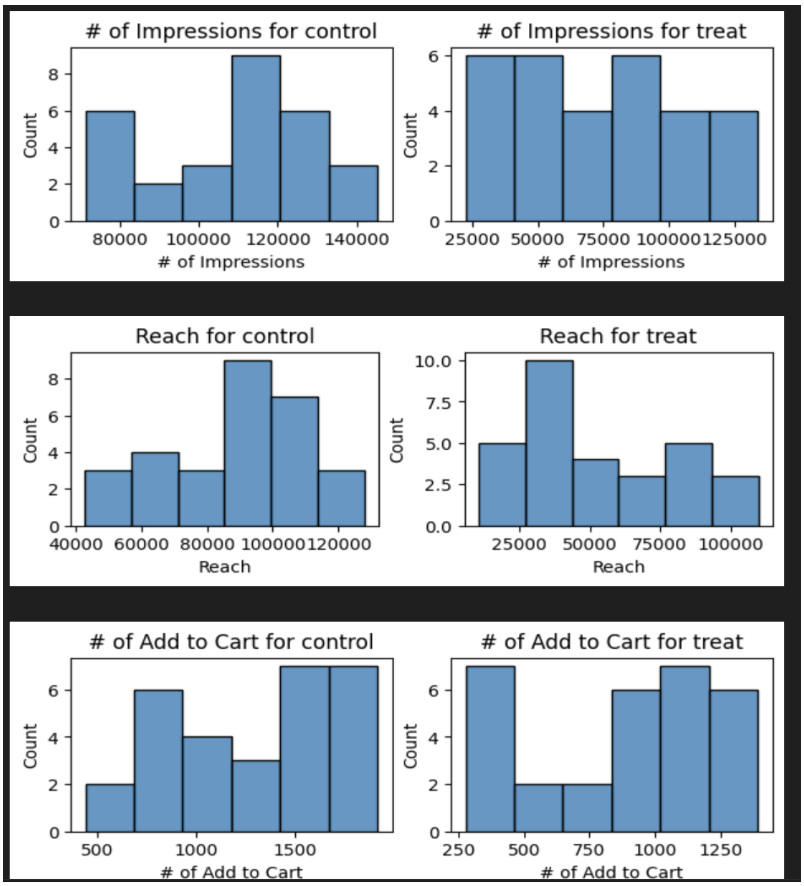

Figure 7: Histogram feature distribution

4.2 Two sample T-test [7]

Another way is to use a two sample t-test to detect this kind of problem. To implement our two sample t-test, we should check if the features are normally distributed and whether there’s a significant difference of variance between control and treatment groups. To check the normality, we use QQ plots as shown below:

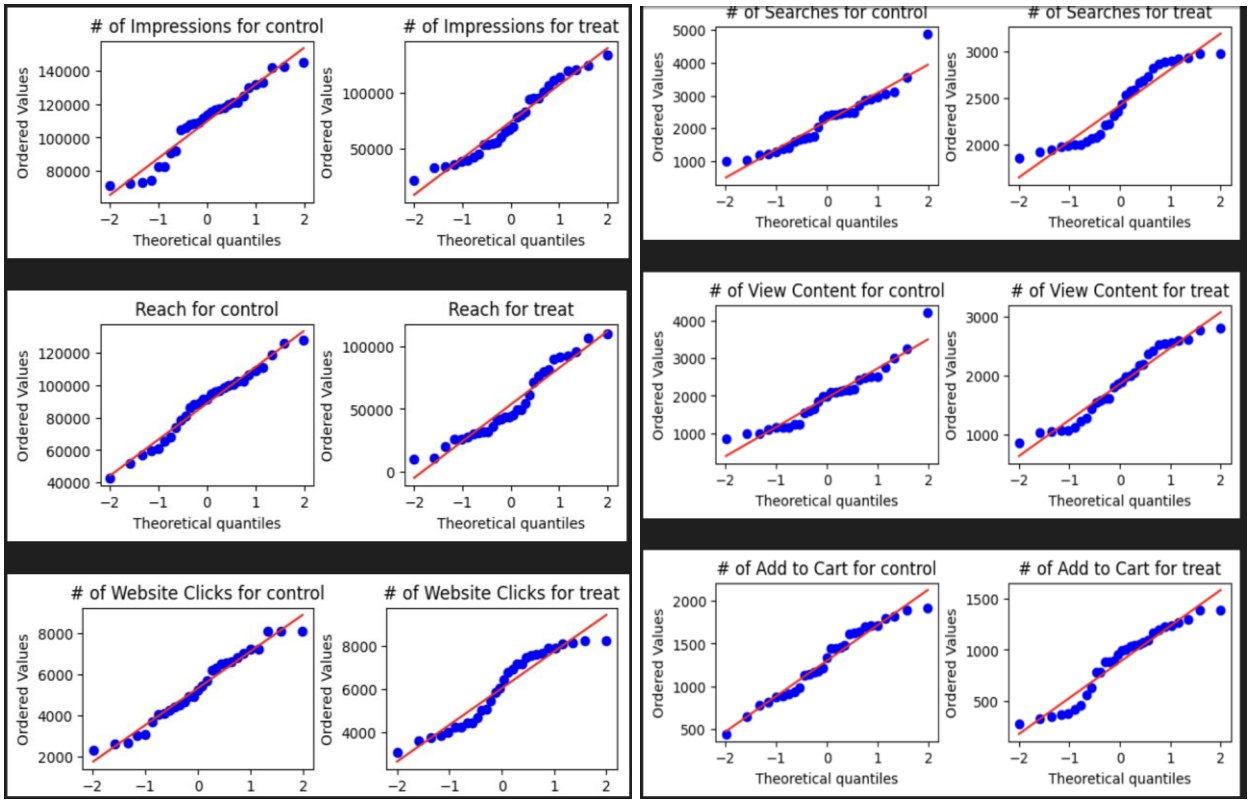

Figure 8: QQ plots to check normality

And here's the difference of variance:

	P-Value for Levene's test
Spend [USD]	0.897559
# of Impressions	0.011125
Reach	0.150998
# of Website Clicks	0.838549
# of Searches	0.003882
# of View Content	0.384204
# of Add to Cart	0.236862

Figure 9: difference of variance

As we can see, almost all features show normal distributions, and using a threshold of 0.001 for p value, there's no significant difference for feature variances between control and treatment groups.

To implement our two sample t-test, we first iterate over the features using a for loop. Within the loop, we skip the 'treatment' column as it is not being compared in the t-test. After that, we create control and treatment groups as follows:

1. For each feature (excluding 'treatment'), the code creates two groups: a control group and a treatment group.
2. The control group consists of values from the 'control' condition (where 'treatment' is 0), and the treatment group consists of values from the 'treatment' condition (where 'treatment' is 1).
3. Missing values are dropped using the dropna() method.

After preparing the data, we can perform two sample t-tests using the "ttest_ind()" function from the "scipy.stats" module, assuming that the variances of the two groups are not equal, and attract the t-statistic and the p-value from test result.

```
features = []
pvalue = []

for feature in data_2.columns:
    # Skip the 'treatment'
    if (feature != 'treatment'):
        control_group = data_2[data_2['treatment'] == 0][feature].dropna()
        treatment_group = data_2[data_2['treatment'] == 1][feature].dropna()

        # Perform the two sample t-test
        t_stat, p_value = ttest_ind(control_group, treatment_group, equal_var=False)

        features.append(feature)
        pvalue.append(p_value)

pd.DataFrame(pvalue, index = features, columns=['P-Value for two sample t-test'])
```

✓ 0.0s

Figure 10: two sample t-test configuration

The result is shown below:

	P-Value for two sample t-test
Spend [USD]	0.004330
# of Impressions	0.000010
Reach	0.000002
# of Website Clicks	0.120549
# of Searches	0.267808
# of View Content	0.637430
# of Add to Cart	0.000087

Figure 11: Result for two sample t-test for continuous features

As we can see, the result from our two sample t-test is similar to what we've found using regression analysis., where "# of Impressions", "Reach", and "# of Add to Cart" show a significant distribution mismatch between control group and treatment group.

However, since both the regression analysis and the two-sample t-tests are parametric measurements. To be specific, we apply the regression analysis based on parameters generated by the GLM model and Wald test, and we apply two sample t-tests based on the assumption that the both control and treatment groups have normal distributions and their variances are equal. What if the distribution of features cannot be captured by the GLM model? Is there any non-parametric test we can use for exploring distribution mismatch?

4.3 Goodness of Fit Chi-Square Test

We've come up with the goodness of fit chi-square test almost immediately. As it performed well for categorical features in our previous dataset, we are also interested in if the goodness of fit chi-squared test applies well for continuous data. The result is shown below.

	P-Value for Chi-square test
Spend [USD]	1.0
# of Impressions	1.0
Reach	1.0
# of Website Clicks	1.0
# of Searches	1.0
# of View Content	1.0
# of Add to Cart	1.0
treatment	1.0

Figure 12: Result of Chi-Square test for continuous features

Unfortunately, using the goodness of fit chi-square test for continuous features would involve binning these features into categories, which could lead to information loss and arbitrary categorization, influencing the test's sensitivity and specificity.

4.4 Kolmorov-Smirnov (K-S) test

Another method for non-parametric tests is the Kolmorov-Smirnov test (KS) test. "Some evidence is presented in this paper indicating that when it is applicable it may be a better all-around test than the chi-square test." [8]. By comparing the empirical distribution function of two groups of data, it can be applied to our continuous dataset easily. Here is the result:

	P-value for K-S test
Spend [USD]	0.070888
# of Impressions	0.000138
Reach	0.000040
# of Website Clicks	0.323709
# of Searches	0.009386
# of View Content	0.966729
# of Add to Cart	0.001349

Figure 13: Result of K-S test for continuous features

The result is similar to the regression analysis. By choosing 0.001 as a threshold as well, “# of Impressions”, “Reach” shows a significant distribution mismatch between control and treatment group. The discrepancy between two methods in terms of "# of Searches" and “# of Add to Cart” could be due to the distribution of this variable being different in a way that the K-S test is sensitive to, but the regression analysis does not find a linear relationship with the treatment when accounting for other factors (There is multicollinearity between features). Also, since our dataset is quite small, the K-S test might not capture the distribution of our features as well as the parametric test as we implemented before.

5. Conclusion

Based on our analysis, we conclude that SRM Regressor Checker demonstrates utility in evaluating both categorical and continuous scenarios. Specifically, two sample t-tests prove effectiveness for analyzing continuous features, while the chi-square test is suitable solely for categorical features and the K-S test is good at capturing distribution of features especially in large datasets (notice that our dataset is not large enough) as a non-parametric method. These findings underscore the importance of selecting appropriate statistical methods tailored to the nature of the data under examination. By leveraging the appropriate techniques, researchers can effectively assess the significance of associations and draw meaningful conclusions from their analyses.

Contribution:

Yuhao Wang: Topic and paper selection, Regression analysis and two sample t-test codings

Jiexin Pan: paper selection for chi-square and K-S test and coding.

Yitong Liu: Report writing

References:

[1] Kang, L. (Spring 2024). Lecture 5 pg.28.

[2] Fabijan, A., Blanarik, T., Caughron, M., Chen, K., Zhang, R., Gustafson, A., Budumuri, V. K., & Hunt, S. (2020, September 15). Diagnosing sample ratio mismatch in A/B testing. Microsoft Research. Retrieved from: https://www.microsoft.com/en-us/research/group/experimentation-platform-exp/articles/diagnosing-sample-ratio-mismatch-in-a-b-testing/

[3] Sajin, S., Zhou, M., Gourishetti, K., Stas Sajin, M. Z., & Gourishetti, K. (2023, October 31). Addressing the challenges of sample ratio mismatch in A/B testing. DoorDash Engineering Blog. Retrieved from: https://doordash.engineering/2023/10/17/addressing-the-challenges-of-sample-ratio-mismatch-in-a-b-testing/

[4] Frequently asked questions. SRM Checker. (n.d.). Retrieved from: https://www.lukasvermeer.nl/srm/docs/faq/#how-can-we-detect-sample-ratio-mismatch

[5] Kohavi, R., & Longbotham, R. (2015). Online controlled experiments and A/B tests. Encyclopedia of machine learning and data mining, 1-11. Retrieved from: https://exp-platform.com/Documents/2023-03-11EncyclopeiaMLDSABTestingFinal.pdf

[6] Kaggle. (n.d.). A/B Testing Dataset. Retrieved from: https://www.kaggle.com/datasets/amirmotefaker/ab-testing-dataset

[7] Nie, K., Zhang, Z., Xu, B., & Yuan, T. (2022). Ensure A/B Test Quality at Scale with Automated Randomization Validation and Sample Ratio Mismatch Detection. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22) (pp. 1-17). ACM. Retrieved from: https://arxiv.org/pdf/2208.07766.pdf

[8] Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. Journal of the American Statistical Association, 46(253), 68–78. Retrieved from: https://luk.tsipil.ugm.ac.id/jurnal/freepdf/2280095Massey-Kolmogorov-SmirnovTestForGoodnessOfFit.pdf