QUESTION 2 – REPORT

Martin Labenne

Answer

Methods

Data Exploration

We are provided with two datasets of features **X1** and **X2** and one response vector **y**. **X1** has 209 samples and 160 columns, no constant features, and no null values. Every value is numeric. In **X1**, every feature is mostly centered on 0 (note the bar at -2e-12, away from the main group) and the variances take a lot of different values varying from 0 to more than 250000 with most features having its variance between 0 and 50000 (*Figure 1*). **X2** has 209 samples and 100 columns, no constant features, and no null values. Every value is numeric. In **X2**, every feature is mostly centered on 0 (note the difference of scale: 1e-13 here vs 1e-12 in X1, the factor of ten should be taken in consideration). The variances take a lot of different values varying from 0 to more than 140000 with most features having its variance between 0 and 50000 (*Figure 2*). **y** has no missing values has a variance of 6523.02 and a mean value of 338.09. No big surprise when visualizing the summary on **(X1, X2)** dataset, there is no big changes in comparison to X1 except of course for the number of variables on screen, we better Standardize the variables anyway.

Model Selection

I used The Lasso since it is a quite simple regularized regression model which allow to perform feature selection using a l_1 norm which leads some regression coefficients of β to be estimated at 0. All non-zero coefficients in β correspond to features that have been selected. With the Lasso, optimal model selection is achieved when the Mean Scared Error is minimized. Thus, the tuning parameter is named λ_{min}. However, it is shown that the selection preformed with λ_{min} is not always particularly great. It generally overfit and too many coefficients are left at small values instead of setting them fully to zero. This improves prediction quality but makes feature selection less valuable. To improve feature selection, we need to choose a bigger λ, only a bit larger than λ_{min}: among all average MSEs (calculated during the cross-validation process to determine λ_{min}) that are at the most one standard error away from the minimal average MSE, the largest one leads to the corresponding hyperparameter λ_{1se}. I made this hyperparameter choice, this way I should have kept a decent prediction quality with a better feature selection.

The Idea was simple: create random bootstrap samples of each dataset I studied, then perform cross validation on the Lasso to estimate the optimal hyperparameter λ_{1se} to achieve a good feature selection. The bootstrap samples aim to estimate the frequency of the selection and the median value of the regression coefficient for each feature. This way, I can justify my feature selection with confidence.

The setup was as follows: I performed a 10-fold cross validation to counterbalance the decrease of precision I imposed to the Lasso (decrease in tolerance, now at 1e-3 instead of default 1e-4, and kept the default max iteration i.e., 1000), this way I can keep precision with an increase of performance on my computer; in addition to this I performed 100 bootstrap samples to achieve decent confidence level while keeping good performance on my computer.

Results and discussions

Figure 4, *Figure 5*, and *Figure 6* show the results of feature selection side by side. The choice of showing only median coefficient in *Figure 5* was made from the intention to keep the results visually simple, the selection frequencies are shown in *Figure 6*. I thought doing a top 20 most frequently selected features was enough to understand what was going on.

Feature selection on X1

After the computation described earlier, I investigated the hyperparameter value during each bootstrap run. On the left panel of Figure 4, it is visible that this study has relatively random hyperparameter selections, suggesting either that penalization does not have a beneficial effect here, or the shapes are not very clearly defined, and they vary substantially from bootstrap run to bootstrap run.

On the upper panel of *Figure 5*, selected features are well spread out along the feature's axis. The median coefficient value can be seen as the expected prediction power of the feature. On the left table of *Figure 6*, I reported the frequency of selection, the median value of regression coefficient and the statistical error of this median across all bootstrap runs.

Feature selection on (X1, X2)

After the computation described earlier, I investigated the hyperparameter value during each bootstrap run. On the right panel of Figure 4, it is visible that this study has relatively random hyperparameter selections (slightly different from those observed with **X1** only), suggesting once again either that penalization does not have a beneficial effect here, or the shapes are not very clearly defined, and they vary substantially from bootstrap run to bootstrap run.

On the lower panel of *Figure 5*, selected features are mainly spread out along X2 features (very few along **X1** and with very low power of prediction). On the right table of *Figure 6*, only 7 features are part on **X1** and 5 of them have less than 1 as absolute median regression coefficient. The selected **X2** features are leading.

Changes between results

From *Figure 7*, features from **X1** do not very correlated to each other, features from **X2** correlate a lot (positively and negatively). Moreover, some of the features from **X1** *do* correlate a lot with features from **X2**.

Since The Lasso will consider each variable separately, it will consider corelated predictors as equally good for prediction and will just "wander" between those features while trying to shrink to zero as many features as possible. That behavior could explain why the features selected from the combined dataset are mainly from **X2** since best features from the combined dataset **(X1, X2)** has bigger absolute coefficient than variables from the first dataset X1 (cf. Figure 6), which means they perform better in explaining the response vector.

To get rid of that bias, I should have used Group Lasso, grouping highly correlated features. That way, it encourages whole groups to be zero or non-zero with similar coefficients.

Figures

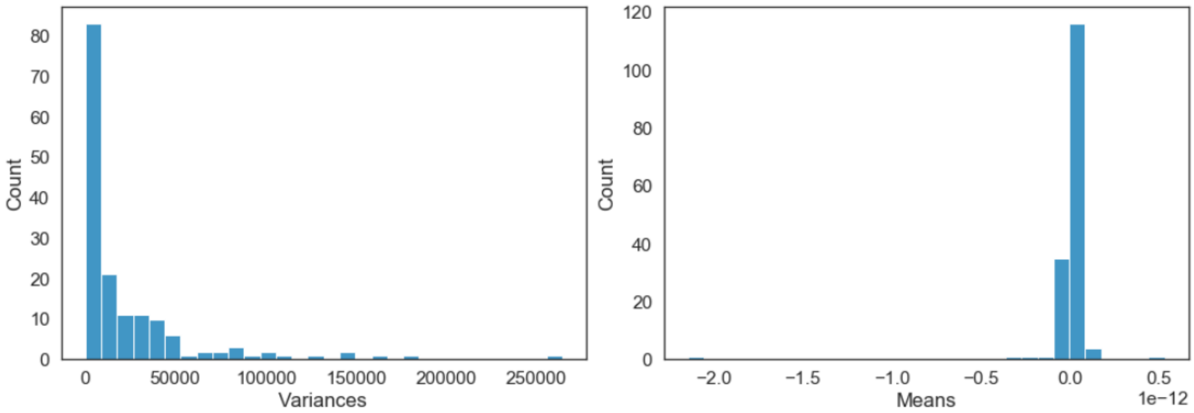


Figure 1: X1, Means and Variances

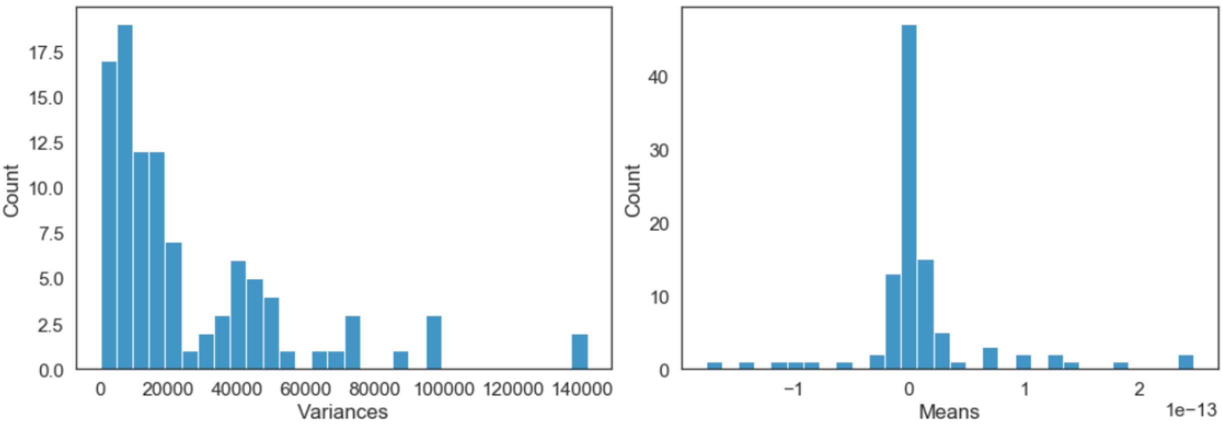


Figure 2: X2, Means and Variances

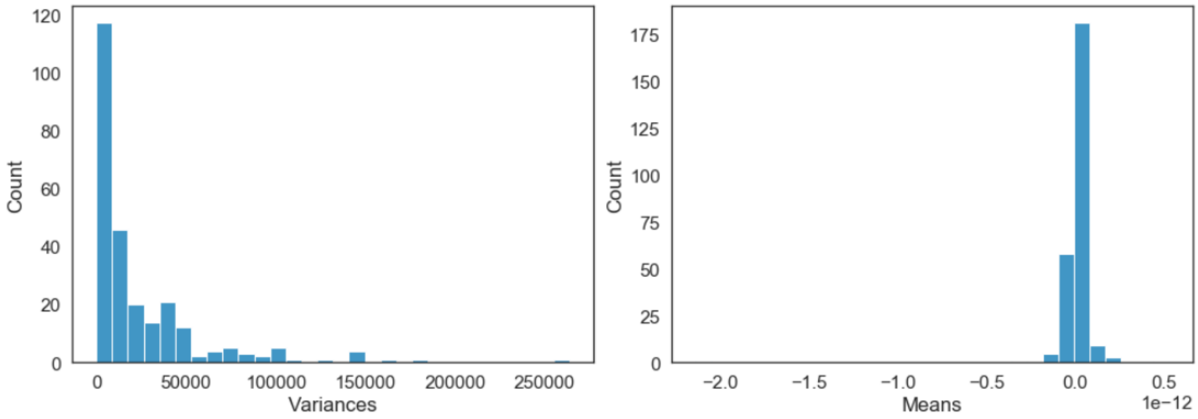


Figure 3: (X1, X2), Means and Variances

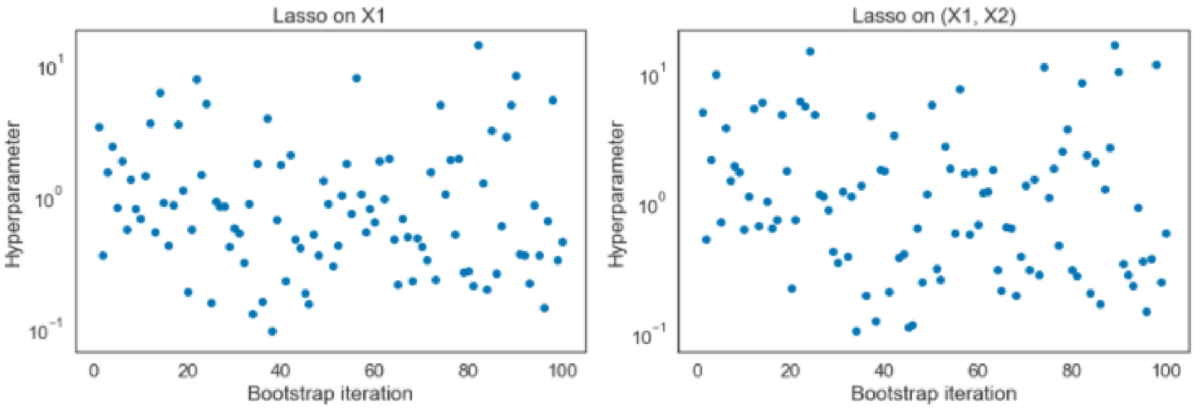


Figure 4: Hyperparameter investigation across all bootstrap runs

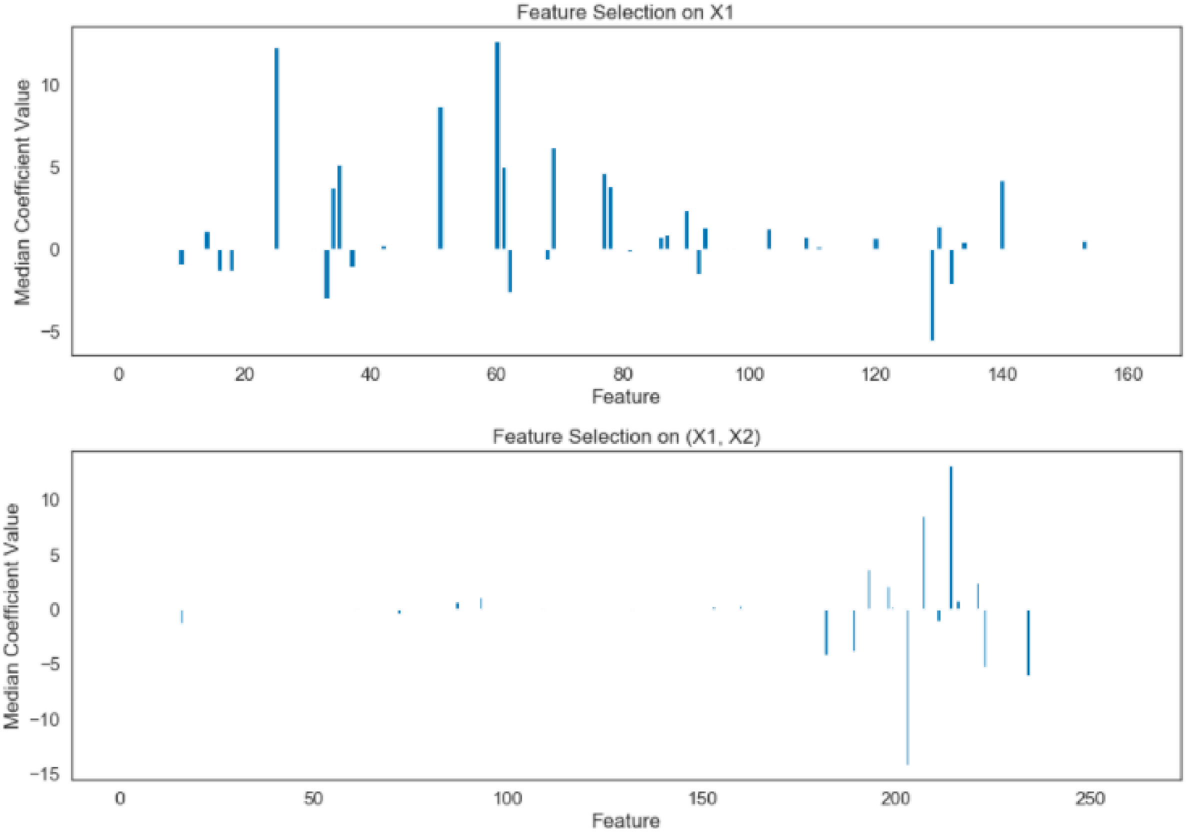
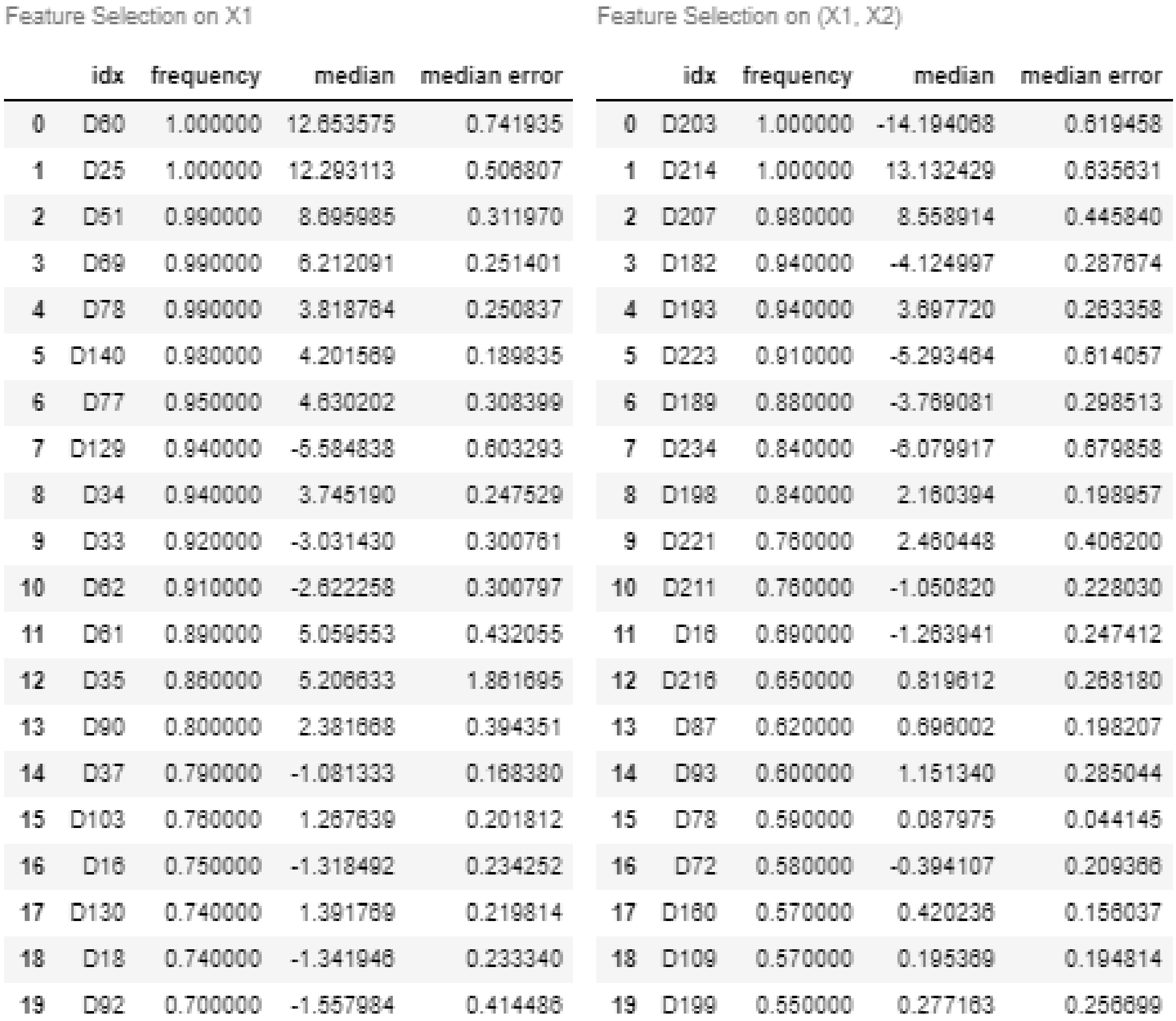


Figure 5: Median values of selected features in both cases



Feature Selection on X1

	idx	frequency	median	median error
0	D60	1.000000	12.653575	0.741935
1	D25	1.000000	12.293113	0.506807
2	D51	0.990000	8.695985	0.311970
3	D69	0.990000	6.212091	0.251401
4	D78	0.990000	3.818764	0.250837
5	D140	0.980000	4.201569	0.189835
6	D77	0.950000	4.630202	0.308399
7	D129	0.940000	-5.584838	0.603293
8	D34	0.940000	3.745190	0.247529
9	D33	0.920000	-3.031430	0.300761
10	D62	0.910000	-2.622258	0.300797
11	D61	0.890000	5.059553	0.432055
12	D35	0.860000	5.206633	1.861695
13	D90	0.800000	2.381668	0.394351
14	D37	0.790000	-1.081333	0.168380
15	D103	0.760000	1.267639	0.201812
16	D16	0.750000	-1.318492	0.234252
17	D130	0.740000	1.391769	0.219814
18	D18	0.740000	-1.341946	0.233340
19	D92	0.700000	-1.557984	0.414486

Feature Selection on (X1, X2)

	idx	frequency	median	median error
0	D203	1.000000	-14.194068	0.619458
1	D214	1.000000	13.132429	0.635631
2	D207	0.980000	8.558914	0.445840
3	D182	0.940000	-4.124997	0.287674
4	D193	0.940000	3.697720	0.263358
5	D223	0.910000	-5.293464	0.614057
6	D189	0.880000	-3.769081	0.298513
7	D234	0.840000	-6.079917	0.679858
8	D198	0.840000	2.160394	0.198957
9	D221	0.760000	2.460448	0.406200
10	D211	0.760000	-1.050820	0.228030
11	D16	0.690000	-1.263941	0.247412
12	D216	0.650000	0.819612	0.268180
13	D87	0.620000	0.696002	0.198207
14	D93	0.600000	1.151340	0.285044
15	D78	0.590000	0.087975	0.044145
16	D72	0.580000	-0.394107	0.209366
17	D160	0.570000	0.420236	0.156037
18	D109	0.570000	0.195369	0.194814
19	D199	0.550000	0.277163	0.256699

Figure 6: Frequency of selection, median coeficient value and median error for the top 20 most frequently selected features in both cases. The tables are ordered by frequency and by absolute median value

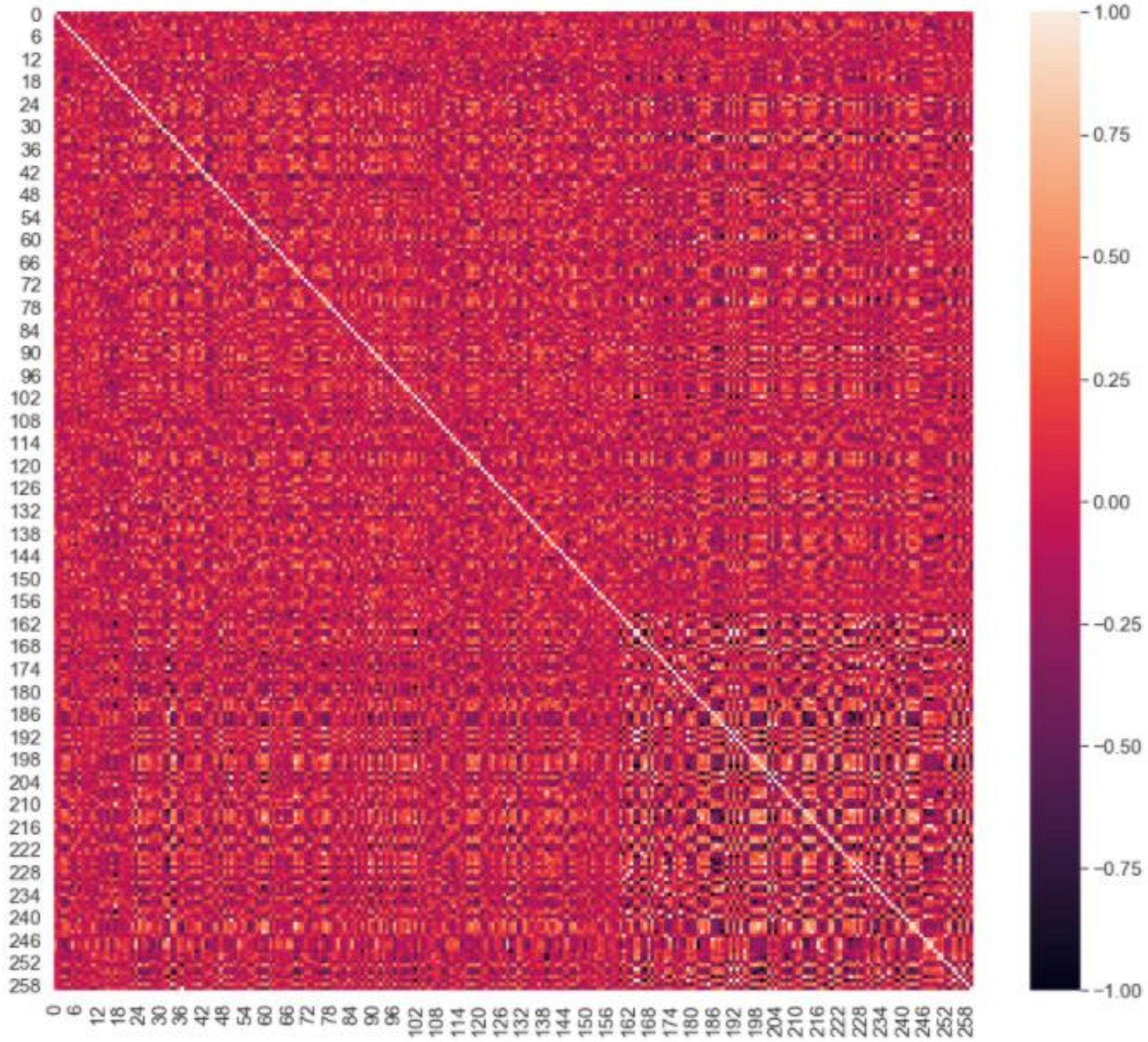


Figure 7: Correlation matrix of (X1, X2). The first 160 features come from X1, the following features come from X2.