Lab Report: Bike-Sharing Data Analysis

MA 575 Fall 2021 - C3 Team #2

Ali Taqi, Hsin-Chang Lin, Huiru Yang, Ryan Mahoney, Yulin Li

12/01/2021

```r
recompute <- F
if(recompute){
  # Obtain the exhaustive dataset of loss values for every unique day ordered pair
  df_loss <- data_2011 %>% get_df_loss
  # Write CSV to avoid future recomputation
  write.csv(df_loss, "df_loss.csv", row.names = F)
} else{
  df_loss <- read.csv("df_loss.csv")
}
```

0.1 Growth Rate Estimation using Environmental Loss Functions

Suppose we have two days (observations) d and d'. Consider the reduced dataset where the observations are solely the vector of our continous variables, in our case, we have three: `atemp`, `hum`, and `windspeed`. Also, let's use `holiday` to focus only on non-holidays. We can adjust for environmental factors but comparing two "environmentally equivalent" days without accounting for `holiday` may very well throw off our estimates of $\hat{g}$. Before we continue, note that we normalize the variables so the loss function is unit-agnostic. So, $d = (x_1, x_2, x_3)$ and $d' = (x'_1, x'_2, x'_3)$ where x_i represents the normalized value of predictor i.

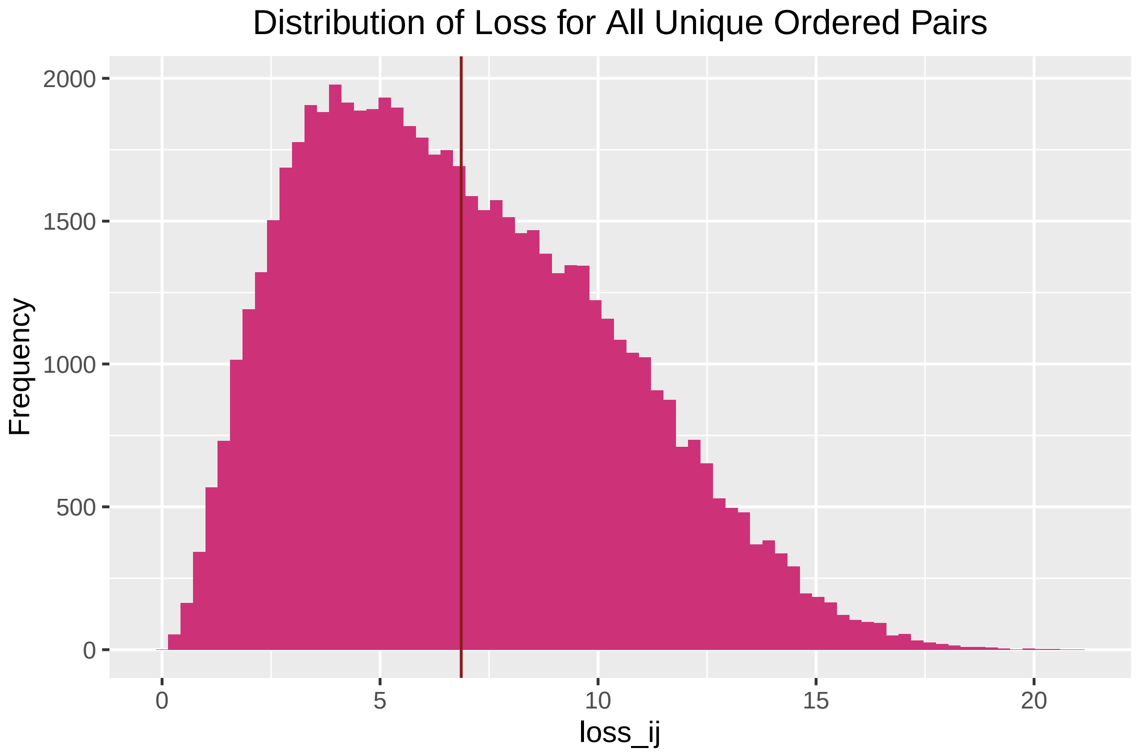
Furthermore, let $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ be a weight vector, more heavily weighing variables deemed more important in determining "environmental similarity". We deem this to be `atemp`. Now, consider the following loss function, which is simply the weighted difference of our normalized predictor variables.

$$\mathcal{L}_d(d') = |\vec{\delta}| \text{ where } \delta_i = \alpha_i(x_i - x'_i)$$

Now, suppose we enumerate all possible unique pairs of days, which we denote $\Pi = \{\pi_{ij} = (d_i, d_j) \mid i < j\}$. Note that we use the lower-triangle scheme, where we choose $i < j$ so that $i - j < 0$. This will prove useful later on since we want day i to preceed day j; this corresponds to the unique pairs found in the indices of a "lower-triangular matrix". In any case, we compute $\mathcal{L}(\pi_{ij})$ for every pair and obtain the set (which we call `df_loss`) $\mathcal{L}(\Pi) = \{l_{ij} = \mathcal{L}(\pi_{ij}) \mid i < j\}$.

Now, here is the magic! We are not interested in taking pairs π_{ij} which are not very environmentally similar. After all, our quest is to estimate growth between days where we in theory expect a similar amount of users by marginalizing environmental factors. So, there comes a question, which pairs do we discount as days too disparate to be considered environmentally equivalent? Well, our good foresight in normalizing the continuous variables and weighing them properly (to the best of our abilities) means that we can consider the distribution of l_{ij}, which we see below:

```r
plot_loss_dist <-
  df_loss %>%
    ggplot() +
    geom_histogram(aes(x = loss_ij), bins = 75, fill = "violetred3") +
    geom_vline(xintercept = mean(df_loss$loss_ij), color = "firebrick4") +
    labs(y = "Frequency", title = "Distribution of Loss for All Unique Ordered Pairs")
plot_loss_dist
```



A quick note: it should not be surprising that we obtain a distribution that resembles a χ^2 distribution. After all, since we assume our predictors are normal, their differences will be too; since our loss is essentially the norm of a multivariate normal, this explains our distribution shape! As such, since it is part of the exponential family, we expect the distribution to be approximtely normal, roughly speaking. So, since we have done a good job by normalizing our variables, a valid strategy to discount pairs/include them is to use a cutoff anywhere below the mean of our distribution μ. We could take for instance, $\mu - 1\sigma$ to be our cutoff for our upper loss bound B.

So, obtain our selected set of pairs of days, cutting off the maximum loss at the selected upper bound B. That way, we obtain the selected pair set by filtering Π with respect to the chosen loss upper bound B:

$$f_{loss}(\Pi, B) = \tilde{\Pi}_B = \{\pi_{ij} \mid l_{ij} = \mathcal{L}(\pi_{ij}) \leq B\}$$

Next, we also filter for what we call an "index difference bound". This is much less critical than the loss bound, as it is only impactful for low values. Namely, the idea is simple: we don't want to pick days too close to each other. We know days close to each other will have similar environmental conditions, so to avoid pesky autocorrelation, we impose a bound on how close the days are. Let $\rho := \min(i - j)$. In our lower-triangle matrix analogy (where we imagine indices as positions of matrix entries), ρ denotes which diagonal band we are considering. In any case, by imposing a lower bound on $i - j$, to pass the index difference filter, every pair must be atleast ρ days apart, or in other words, be at least ρ diagonal bands close to the main diagonal. In practice, we find low values of ρ to produce the most impactful filters, this makes sense, owing to the autocorrelation observed in days close to each other mentioned above.

$$f_{idx\_diff}(\Pi, \rho) = \tilde{\Pi}_\rho = \{\pi_{ij} \mid i - j > \rho\}$$

Finally... we compute the estimate derived from the filtered pair set $\tilde{\Pi}$ as follows. Let c_i denote the count of bike users in day i, and c_j those in day j. **Since we have, in theory, marginalized out the environmental factors (and holidays), we may obtain a reasonable estimate for g by taking the observed growth observed between day i and day j, then taking the average over all the pairs.**

Since by construction, we used lower-triangular indices, this means day j is always after day i. As such, we can imagine each pair, which has marginalized environmental factors, to contain true information about user-base growth since day j is roughly equivalent to day i in environmental factors and is in the future of day i! In other words, take the set of estimates for every $\pi_{ij} \in \tilde{\Pi}_{B,\rho}$ and $\hat{\gamma}_{ij} := \frac{c_j}{c_i}$:

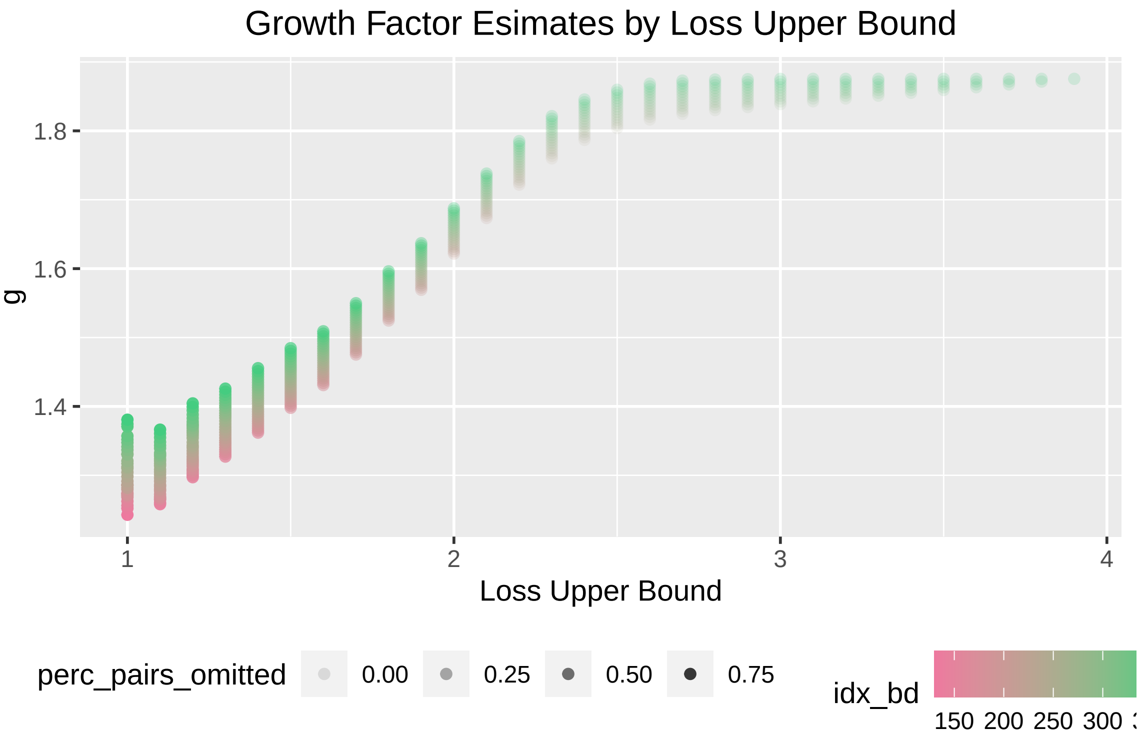
$$\mathbb{E}[\{\gamma_{ij}\}] := \hat{g}$$

0.2 Estimate Performance over Bound Paramater Space

```r
# Set a sequence of index bounds
idx_bds <- seq(122, 365, 8)
# Set a sequence of loss bounds: more critical
loss_bds <- seq(1, 4, 0.10)
# Obtain the parameter dataframe of g estimates
df_param <- df_loss %>% get_df_param(idx_bds, loss_bds)
# Peek at df_param
head(df_param)
```

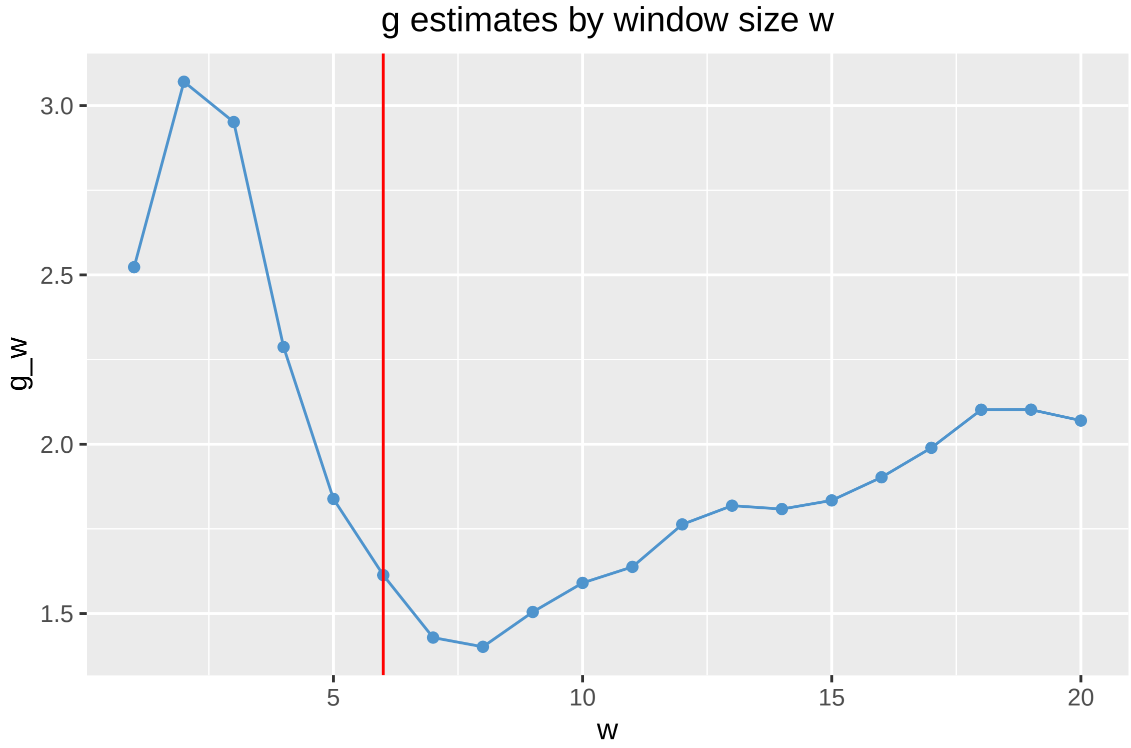
```
##   idx_bd loss_bd        g    n
## 1    130     1.0 1.242667  521
## 2    138     1.0 1.252098  506
## 3    138     1.1 1.258080 3198
## 4    146     1.0 1.256074  497
## 5    146     1.1 1.260398 3149
## 6    146     1.2 1.297205 8044
```

```r
# Plot the estimates over bound parameter space
df_param %>% g_plot()
```



0.3 Window Technique

```
# Obtain the table of g esimates by window paramater w
tbl_window <- purrr::map_dfr(1:20, function(w){data.frame(w = w, g_w = window_g(w))})
# Plot the values
plot_w <- plot_window(tbl_window)
plot_w
```


g estimates by window size w
g_w
w