**Data limitation**: the data collection covers a short time window, which is not sufficient to be utilized to re-construct the traffic pattern for an IP address block.

**Assumption:**

1. IP address bock reservation scheme:
   * xxx.xxx.-.- style IP address reservation: (e.g. all ip address starting with 123.234 are reserved for a particular organization and considered as hotspot)
   * [xxx.xxx.xxx.- , xxx.xxx.xxx.-] style IP address reservation: (e.g. all ip addresses falling on the range between 123.234.115.001 – 123.234.120.254 are reserved and considered as hotspot)
2. Unique user identifier:
   * Ip\_address + user\_agent\_str is a proxy for unique user identifier

An IP address could be assigned to different users at different time point. However, during a day, it is also possible that a single user could be assigned with multiple IP addresses if dynamic scheme is adopted after rebooting system.

Limitation:

1. This proxy could overstate the number of unique users, with respect to some user could surfing internet with computer and mobile devices simultaneously. In this situation, using ip\_address + user\_agent\_str as proxy for unique user identifier could overstate the number of unique users;
2. In some organization, multiple user scould use same public IP address. At this situation, using ip\_address + user\_agent\_str could understate the number of unique users. Therefore, utilizing the internet visiting patterns regarding the domain name could be capable of separating individuals (In this study, the approach coping with this situation is not discussed).
3. OS distributions of user computers within a IP address block:

Within a cooperate organization, it is very likely that the distribution of Computer OSs concentrates on a particular type of OSs system. However, the concentration or diversity of OSs could vary according to the business and culture of the corporate. And, it is also possible for some IT service company’s OS distributions could be as diversified as the nation-wide OS distribution. In contrast, within a public hotspot, e.g. Starbucks, the OS distribution of internet terminals supposed to resonate the benchmark OS distribution within its geographic area (Be caution, customer bases highly focusing on segment of residents. Therefore, such claim may not be true for them).

1. Daily traffic pattern and Weekly traffic pattern:

The internet surfing traffic within an organization should align with work hour schedule or reveal some opposite pattern. Using an industrial cooperate as an example, during weekdays, the traffic should be at high-level during work hours. And, in contrast, it stays at low-level during non-work hours. At weekends, the traffic remains at relatively low level in general. For non-cooperate organization, the traffic could reveal different patterns. For example, Starbuck chain store could generate high traffic beyond cooperate work hours (7:30AM – 8:30 AM, 5:00PM – 7:00PM during weekdays). And daily total users could be at highest level for weekend days.

In conclusion, a hotspot should reveal a regular pattern with respects to users profile, visits information and the nature of the organization.

**PART I: Data preparation:**

In this section, the data process, including feature extraction and transformation, are discussed in details.

Raw data (time\_stamp, ip\_address, uri, complete uri, user\_agent\_str)

**Feature Extraction:**

1. Extract user\_agent\_str, to extract OS information and encode it with a set of binary variables: is\_windows, is\_mac, is\_linux, is\_mobile (Android + iOS)
2. Re-encode time\_stamp with their weekday (1, 2, … and 7) and hours
3. Decompose IP(ipv4) address into 4 components.

**Aggregate Information:**

1. Grouping records/rows by first two components of IP address, weekday, hour
2. Computing aggregate information at hourly level:
   1. #unique\_user: the number of unique combinations of IP address and user\_agent\_str
   2. #windows, #mac, #linux, #mobile
3. Computing aggregate information to ip\_address block (first components of IP address) over a week
   1. Compute windows\_pct, mac\_pct, linux\_pct, mobile\_pct of entire weekly data:

(e.g. windows\_pct = #windows / sum(#windows, #mac, #linux, #mobile) )

* + 1. **dominating\_os\_pct:**  max(windows\_pct, mac\_pct, linux\_pct, mobile\_pct)
    2. **inequality\_idx:** sum(windows\_pct^2, mac\_pct^2, linux\_pct^2, mobile\_pct^2): the metric would emphasize the unevenness of OS distribution
  1. Compute the daily total unique user over a week

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ip\_block (first 2 components) | mon\_user\_total | tue\_user\_total | … | sat\_user\_total | sun\_user\_total |
| 123.245 | 2399 | 2194 |  | 200 | 181 |

* 1. Compute workhour\_pattern\_ind, weekday\_high\_ind and weekend\_high\_ind,
     1. workhour\_pattern\_ind: create a binary variable based on the hourly aggregate, **the value equals to 1** if the distribution hourly user total are concentrated within a window of consecutive hours (de-localization, no necessary to convert time stamp to local time zone); **Otherwise,** **0** if hourly user totals are evenly distributed over a day;

**Calculation demonstration (for an IP address block):**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Hour | 00:00AM | … | 7:00AM | 8:00AM | 9:00AM | … | 4:00PM | 5:00PM | 6:00PM | … |
| hourly\_user\_total | 15 |  | 30 | 120 | 200 |  | 320 | 200 | 45 |  |
| user\_total\_change\_pct\* | - |  | 10% | 300% | 66.67% |  | 10% | 60% | 344% |  |
| Jump\_ind | - |  | 0 | 1 | 0 |  | 0 | 0 | 1 |  |

**Formula:**

user\_total\_change\_pct = abs( #user\_(t+1) - #user\_(t) ) / min(#user\_(t+1), #user\_(t))

**Return:**

Workhou\_pattern\_ind = 1, if sum(jump\_ind over t) >= 2; Otherwise, 0

* + 1. **weekday\_high\_ind:** based on daily user total, **equals to 1** if the average of daily user total for weekdays is substantially larger than the counterpart of weekends; Otherwise, it is 0;
    2. **weekend\_high\_ind:** based on daily user total, equals to 1 if the average of daily user total for weekdays is substantially larger than the counterpart of weekdays; Otherwise, it is 0:

(For ip address reveals no weekday effects, it means that the daily user totals are evenly distributed. In this situation, weekday\_high\_ind = 0 and weekend\_high\_ind)

* + 1. **weekday\_effect\_ind**: **weekday\_high\_ind** or **weekend\_high\_ind**

**Final Aggregate Profile of an IP Address block:**

1. Row identifier**: ip\_address\_block** (e.g. 123.201)
2. Columns about user counts: **average\_user\_total**, weekday\_daily\_user\_total, weekend\_daily\_user\_total
3. Columns about OS distribution: **dominating\_os\_pct, os\_gini\_coefficient (to depict the overall dispersenty** , windows\_os\_pct, mac\_os\_pct, linux\_oc\_pct, mobile\_os\_pct
4. Columns about traffic distribution: **workhour\_pattern\_ind,** **weekday\_effect\_ind**, weekday\_high\_ind, weekend\_high\_ind, weekday

**PART II: Hot-Spot detection:**

**Assumption:**

1. xxx.xxx.-.- : style 1 IP address block;
2. [xxx.xxx.xxx.- , xxx.xxx.xxx.-]: style 2 IP address block;

**Step01:**

Categorize IP address blocks of grouping by 1st and 2nd components of IP addresses into either a) style 1 IP address block or 2) possible style 2 address block (the first 2 components ip address can be fully split into multiple style-2 ip address blocks).

Categorization rule based on the **daily average user**:

1. If daily average user is very large: Possible style 2 address blocks, the part of addresses are subject to re-assignment and other part of addresses reserved for organizations. Therefore, as whole address block, it could include a huge number of unique users and large diversity of OSs. This address blocks (xxx.xxx.-.-) is recommended to fully split to (xxx.xxx.xxx.- , xxx.xxx.xxx.-). And sub-blocks should be used against to aggregate data in the same fashion as described above to evaluate if the sub-block can be considered as a hotspot.
2. If daily average user is relatively large, it could be style 1 ip address block and should be evaluated by processes described in the step 02.

**Step02: training classifier for**

1. Frame the study as non-supervise learning, using the aggregate data described above to do clustering analysis or naïve bayes decision to segment the ip address block.
2. Frame the study as semi-supervise learning, if some ip blocks’ hot-spot status can be known. Using this given label to train a classifier.