Università Cattolica Sacro Cuore

Statistical and Actuarial Sciences

mj: Data Business Analytics

Spatial Machine Learning modelling: End-to-End web App solution

Supervisor:
Dr. Marco DellaVedova

Author:
Niccolò Salvini

Assistant Supervisor:
Dr. Vincenzo Nardelli

AY 2019 / 2020

Spatial Machine Learning modelling: End-to-End web app solution

Niccolò Salvini[1]

date: Last compiled on 20 settembre, 2020

[1] https://niccolosalvini.netlify.app/

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Main themes:

- (open data)
- research question
- milan market real estate
- latest improvements in the subject matter
- why both bayesian and non bayes methods
-

We are living in the big data era, so we could be brought to think that everything is a “one click” distant from us. Well, this is not totally true, moreover in some places this is truer. The main issue can be addressed to the lack of open data and the lack of relative infrastructure. This settings characterizes slow old economies and unfortunately Italy is one of them. Economies, and citizens on a later step, can largely benefit from public data and its usage. Some people in addition are in favor of the position that all data should be open. Since I am living in italy and my (Lovelace et al., 2019) goal is to an alyse market

The importance of data indeed justifies its accumulation and according to the latest reports is surpassing gold, despite these periods of uncertainty. The expression data is the new oil has never been so appropriate in these times. On the other hand is not for sure easy to assign a price amount to data due to its untangible nature. the most straightforward and liberal approach could lead us to think that the price data should be exchanged the piceThe value attributed to data is not for sure selrmarkdown::pandoc_available("1.2") f explanatory. It really depends on two major metrics: the usage that can be done through (with respect of the state of the art technology) it and the functionality with respect of other existing data. some data can be strategically important given the fact that someone already possess the complementary and can attribute some sort of competitive advantage. On the other hand as already been highlighted it really depends on the existing technology stack. Some data can be very useful but too costly either to process or to store.

$$f\left(k\right)=\binom{n}{k}p^{k}\left(1-p\right)^{n-k} \tag{1.1}$$

During an interesting conversation with some friends we had a discussion on how data should be treated: as a sort of currency or a sort of commodity (raw material). some people may say that the inner functioning is pretty much as a commodity. It gains value by its specialized usage and treatment. Sometimes a collection of data can represent the complementary part of a more general dataset that can not be used otherwise, in this analogy case a semi-finished product. commodities sometimes are calmed, so that their prices are fixed to a certain amount, so it is for data, the

Durante una conversazione con alcuni amici[1]

La ricerca che ho inteso fare sul mercato degli affitti a Milano mi ha aperto le porte a comprendere come poco digitalizzata e all'avanguardia sia la nostra amata penisola. L'indisposizione ai dati aperti, coperta da un sottile velo di ipocrisia chiamato privacy (ma quale?), ha reso non solo impossibile reperire

[1] footnote a caso

i dati geospaziali tramite API di alcune aree dell'italia, ma ha reso necessario che costruissi delle funzioni che li estressero, appoggiandomi a cavilli. La questione è legale e relativaemnte complessa, e di certo la tesi non si indirizza a questi problemi, ma i dati che sono stato capace di scoprire e di farlo nella assoluta legalità si appoggiano ad una mancanza di autorizzazioni al trattamento che Immobiliare.it ha nel suo sito. Un altro esempio di ritardo tecnologico riguarda l'assenza dei dati di elevazione su alcuni territori italiani. Se per esempio immobilaire, come ha fatto un altro grosso player sul mercato, avesse apposto una checkbox obbigatoria da contrassegnare con relativi termini e trattamento dei dati io non avrei potuto accedere ai dati. La situazione aldifuori dell'italia è abbastanza uniforme, eccetto qualche paese noto come Germania e Francia, e meno noto come la Polonia, con tutte altre piattaforme e regole di trattamento dati. La domanda quindi sorge spontanea, perchè i dati degli italiani e degli europei sono meno accessibili dei dati degli americani? Mentre in America è sufficiente richiamare un API con latitudine e longitudine della quadrettatura di terra necessaria per ottenere i dati[2] di elevazione (.tif), in Italia l'unica soluzione è pagare google che tramite le sue private API è in grado di venderceli dietro autenticazione. La risposta aldilà dei confini della legge presumibilmente risiede in un congiunturale ritardo di infrastrutture tecnologiche condivise e di indirizzo comune europeo sulla questione. L'esigenza di dati aperti nasce per la risoluzione di problemi comuni a tutti, i dati sanitari hanno la missione di tentare risolvere problemi di natura sanitaria, i dati economici auspicabilmente curano probelmi o asimmentrie di un mercato. Il mercato degli affitti a Milano gode di sempiteerna gloria e ha visto la crescita degli affitti e dei prezzi degli immobili di paripasso al punto che una bolla è stata presunta. Diversi fattori hanno reso tale il fenomeno e diverse opinioni si sono spese sul tema. Alcuni pensano che dopo Expo la città abbia goduto di una spinta economia e innovativa che l'ha resa un'isola felice in mezzo ad un'italia che affanna. Altri ritengono che Milano goda di ottime infrastrutture, ma che la sua notorietà ed il suo appeal si sia sostituito a tutto

[2] https://it.wikipedia.org/wiki/Dato

quello che manca nelle altre città, ma che in Milano appare. La mia opinione è che sia una media di questi due pareri. Un altro fattore è imporatante nella descrizione del fenomeno: l'asimmetria di infromazione tra chi cerca casa a Milano venendo da fuori e colui che affitta. Tale asimmetria viene ancora più esasperata al crescere della fretta che l'entrante ha nel trovare la locazione opportuna. La scelta diventa in molti casi antieconomica, nello specifico la domanda si genuflette all'offerta e accetta le svattaggiose condizioni proposte. Infatti quello che appare certo è che i prezzi degli affitti se comparati ai salari per posizioni junior e di stage è falsato. Proprio qui nasce l'esigenza di approfondirne il perchè e fornire all'utente finale (un potenziale studente, un futuro lavoratore etc.) uno strumento che gli permetta di capire il prezzo stimato tramite predizione spaziale date le coordinate geografiche e gli attributi dell'appartamento e contestualmente fornire un mezzo di comparazione per altri immobili nelle vicinanze. Dall'altro lato dia un'idea chiara a chi vuole dare in affitto l'immobile, un prezzo rappresentativo, che ha fondamento nel modello utilizzato e nelle assunzioni che lo stesso modello impone alla realtà. Questo fa sì che da entrambi i lati ci sia trasparenza e che eventuali maggiorazioni di prezzo richiesto rispetto al sopradetto modello vengano penalizzate in favore di sconti applicati su altri immobili. Auspicabilmente i prezzi già gonfi si smusseranno in tutta la regione spaziale considerata, adattandosi alla domanda piuttosto che al capriccio dell'offerta.

Chapter 2

Scraping

2.1 What is Scraping

Lo web scraping è una tecnica di estrazione dei dati da pagine internet statiche o dinamiche in maniera automatica e simulatanea (Wikipedia, 2020). L'impossibilità di reperire dati aperti aggiornati riguardo l'affitto sul mercato italiano mi ha spinto a sviluppare sofisticate tecniche di estrazione di dati orientate ad alleggerire lo sforzo e aumentare la velocità di reperimento: da una parte nel preprocessing del dataset, nella successiva del fragente del modelling, per finire con la reattività di risposta dell'applicazione. Le informazioni sui siti appaiono spesso ordinate e semplici, tuttavia ogni sito web ha una propria architettura e un proprio linguaggio. Per architettura intendo struttura gerarchica secondo cui è organizatto un sito internet: una semplificazione della struttura di un sito web può essere un insieme di cartelle innestate una dentro l'altra collegate tra loro da riferimenti tramite l'url. la natura gerarchica della struttura prevede che si usi un linguaggio che fa propria questa caratteristica, HTML è il preferito. L'html si organizza in nodi ed angoli, esattamente come un grafo; che aggiunta la componente gerarchica fa sì che questo sia un albero. Difatti spesso ci si riferisce alla struttura delle pagine web come html tree. Ogni elemento nella pagina ha un suo preciso posto nel codice sorgente della stessa e ha un preciso valore o più valori.

Possiamo immaginare ogni nodo della pagina come una lista di valori che è collegata ad un nodo precedente detto padre da una struttura gerarchica superiore, ed eventiualmente ad un nodo successivo detto figlio. Pertanto tutte le informazioni che giacciono sotto al nodo padre sono parenti del nodo padre e sono direttamente collegate (directed nel senso dell'interpretazione), parallelamente ci saranno altri nodi padre che saranno adiacenti al nodo padre, i quali avranno nodi figli e così via. La complessità della pagina e del codice è tanto maggiore quanto il livello dell'albero aumenta, tanto più l'albero è folto tanto più sarà difficile individurare il ramo o la foglia che ci interessa. Ragionevolemnte accade lo stesso per la funzione di scraping e il tempo di scraping. Html organizza i contenuti e le relazioni tra loro, il css (Cascading Style Sheets) invece si occupa dello stile e della formattazione degli stessi. il css è uno strumento molto potente in mano ad uno scraper perchè permette di recuperare informazioni simili tra loro ma che occupano nodi con posizione gerarchica diversa all'interno della pagina. Pertanto una volta letto l'html della pagina sarà necessario recuperare la query css per raccogliere tutti gli elementi di interesse tramite la funzione di scraping. Successivamente occorre notare che l'encoding da html a stringa di testo non è quasi mai lineare, spesso occorre riformattare, cancellare spazi, convertire la natura dell'oggetto estratto etc. Il successivo elemento di complessità incontrato durante questa prima fase è stato interfacciarsi con un server attento alle richieste GET degli utenti. I dati viaggiano in pacchetti da un server che ospita un sito internet al nostro laptop. tutte le volte che cerchiamo di accedere ad un sito stiamo mandando una richiesta di ricezione di pacchetti dati ad un server in qualche luogo remoto del mondo. Quandi bussiamo alla porta del server se non siamo sospetti e superiamo i criteri autostabiliti dal server questo risponde, e lo fa con un numero che spazia da 200 a 500, due esempi: 200 se la risposta è positiva, 404 se la risposta è negativa. I criteri secondo cui gli utenti sono calssificati secondo utente normale o utente sospetto (aka bot) sono sintetizzati in un documento di testo chiamato robot.txt. Questo file di testo raccoglie tra le altre due infromazioni principali il delay time, cioè il tempo preferito dal server che deve intercorrere tra una richiesta dati e la successiva e quale utente è autorizzato ad accedere. Ogni utente posside un indirizzo IP che nelle richieste a server si codifica in user agent, cioè una stringa di testo dove vengono raccolte le infromzioni significative circa il dispositivo da cui provengono le richieste, un esempio:

‘Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36’,

dove ogni segmento della stringa rispecchia una caratteristica del laptop del richiedente, Chrome/54.0.2840.71 è la versione del browser chrome da cui proviene la richiesta Safari/537.36’, è il motore di ricerca etc.

2.2 Scraping Best Practices and Robot.txt

Robots.txt files are (rivedi citation) a way to kindly ask webbots, spiders, crawlers, wanderers and the like to access or not access certain parts of a webpage. The de facto ‘standard’ never made it beyond a informal “Network Working Group INTERNET DRAFT”. Nonetheless, the use of robots.txt files is widespread (e.g. https://en.wikipedia.org/robots.txt, https://www.google.com/robots.txt) and bots from Google, Yahoo and the like will adhere to the rules defined in robots.txt files, although their *interpretation* of those rules might differ.

Robots.txt files are plain text and always found at the root of a website’s domain. The syntax of the files in essence follows a fieldname: value scheme with optional preceding user-agent: ... lines to indicate the scope of the following rule block. Blocks are separated by blank lines and the omission of a user-agent field (which directly corresponds to the HTTP user-agent field) is seen as referring to all bots. # serves to comment lines and parts of lines. Everything after # until the end of line is regarded a comment. Possible field names are: user-agent, disallow, allow, crawl-delay, sitemap, and host. For further notions (Meissner and Ren, 2020, goo (2020))

Some interpretation problems:

- finding no robots.txt file at the server (e.g. HTTP status code 404) implies that everything is allowed
- subdomains should have there own robots.txt file if not it is assumed that everything is allowed
- redirects involving protocol changes - e.g. upgrading from http to https - are followed and considered no domain or subdomain change - so whatever is found at the end of the redirect is considered to be the - robots.txt file for the original domain
- redirects from subdomain www to the doamin is considered no domain change - so whatever is found at the end of the redirect is considered to be the robots.txt file for the subdomain originally requested

For the thesis purposes it has been designed a dedicated function to inspect whether the domain requires specific actions or prevents some activity on thw target website. The following `checkpermission()` function has been integrated inside the scraping architecture and it is called once at the very beginning.

```r
library(robotstxt)
dominio = "immobiliare.it"

checkpermission = function(dom) {

    robot = robotstxt(domain = dom)
    vd = robot$check()[1]
    if (vd) {
        cat("\nrobot.txt for", dom, "is okay with scraping!")
    } else {
        cat("\nrobot.txt does not like what you're doing")
        ## stop()
    }
}
checkpermission(dominio)
```

```
##
## robot.txt for immobiliare.it is okay with scraping!
```

Further improvements in this direction came from the `polite` package (Perepolkin, 2019) which combines the power of the `robotstxt`, the `ratelimitr` to rate-limiting requests and the `memoise` for response caching. This package is wrapped up around 3 simple but effective ideas:

> The three pillars of a polite session are seeking permission, taking slowly and never asking twice.

The three pillars constitute the Ethical web scraping manifesto (Densmore, 2019) which are common shared practises that are aimed to self regularize scrapers. This has not nothing to do with law but since many scrapers themselves, as website administrators or analyst, have fought with bots. Bots might fake out real client logs and might stain analytics, so here it is born the choice to fine common ground and politely ask for permission.

2.3 User agents, Proxies, Handlers

Everytime a user enters a website what he is really doing is sending an HTTP request to the website server with some information packed. This can be easily thought as a generic person A that rings the door's bell of person B's house. A comes to the B door with its personal information, its name, surname, where he lives etc. At this point B may either answer to A requests by opening the door and let him enter given the set of information he has, or it may not since B is not sure of the real intentions of A. This typical everyday situation in nothing more what happens billions of times on the internet everyday, the user (in the example above A) is interacting with a server website (part B) sending packets of information. If a server does not trust the information provided by the user, if the requests are too many, if the requests seems to be scheduled due to fixed sleeping time, a server can block the requests. In certain cases it can even forbid the user to be on the website. The language the two parties talks are coded in numbers that ranges from 100 to 511, each of which ha its own significance. A popular case of this type of interaction occurs when users are not connected to internet so the server responds 404, page not found. Servers are built with a immune-system like software that raises barriers and block users to prevent dossing or other illegal practices.

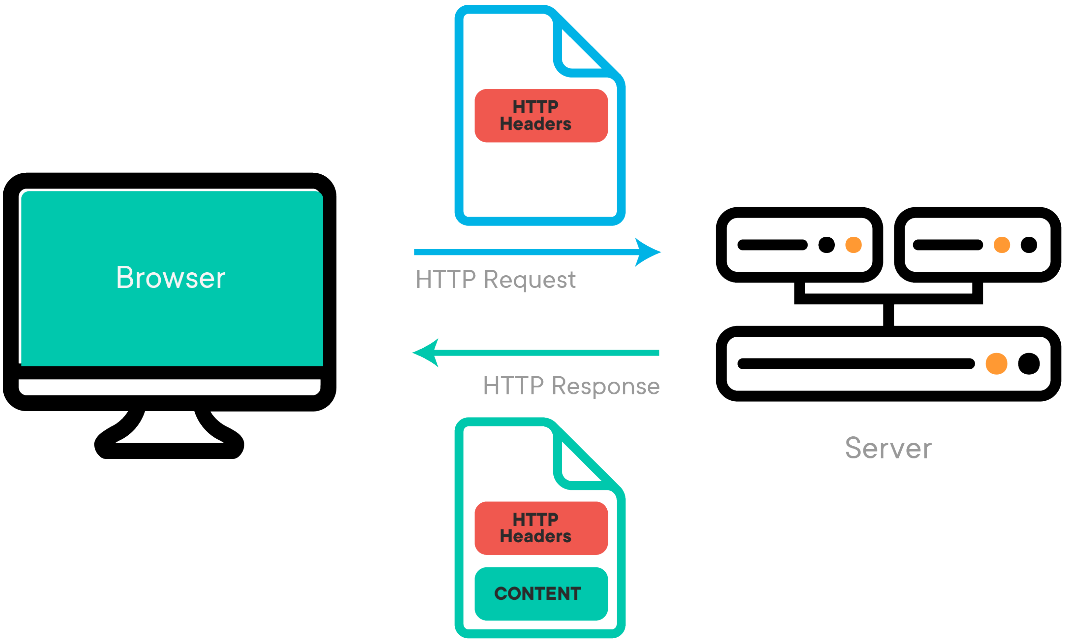


Figure 2.1: How Web Works

This procedure is a daily issue to people that are trying to collect information from websites. Google does it everyday with its spider crawlers, which are very sophisticated bots that performs scarping over a enormous range of websites. This challenge can be addressed in multiple ways, there are some specific Python packages that overcome this issue. The are also certain types of scraping as the Selenium web driver automation that simulates browser automation. Selenium allows the user not to be easily detected by the server immune system and peaceful. In here precautions have not been taken lightly, and a simple but effective approach is proposed.

2.3.1 User agents Spoofing

A user agent (who, 2020) is a string of characters in each browser that serves as an identification agent. The user agent permits the web server to be able to identify the user operating system and the browser. Then, the web server uses the exchanged information to determine what content should be presented to particular operating systems and web browsers on a series of devices. The user agent string contains the user application or software, the operating system (and their versions), the web client, the web client's version, and the web engine responsible for the content display (such as AppleWebKit). The user agent string is sent in form of a HTTP request header. Since the User Agents acts as middle man between the client request and the server response what it would be better doing is to actively faking it so that each time a web browser presents himself to a web server it has a different specifications, different web client, different operating system and so on.

The simple approach followed was building a vector of samples of different existing and updated User Agents (UA). Then whenever a request from a browser is served to a web server 1 random string is drawn from the user agents pool. So each time the user is sending the requests it appears to have a different "identity". Below the user agents rotating pool:

```r
set.seed(27)
agents = c("Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/602.2.14 (KHTML, like Gecko) Version/10.0.1 Safari/602.2.14",
    "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.98 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.98 Safari/537.36",
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36",
    "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0")
agents[sample(1)]
```

```
## [1] "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36"
```

An improvement to this could be using also rotating proxies. A proxy server acts as a gateway between the user and the server. It's an intermediary server himself, separating end clients from the websites they are browsing. Proxy servers provide varying layers of functionality, security, and privacy are some of the examples. While the user is exploiting a proxy server, internet traffic flows through the proxy server on its way to the server you requested. The request then comes back through that same proxy server and then the proxy server forwards the data received from the website to you. Many proxy servers are offered in a paid version so in this case since security barriers of the target website are not high they will not be implemented. It has to be mentioned that many online services are providing free proxies but the turnaround of this solutions are many, two of them are: - Proxies to be free are widely shared among people, so as long as someone has used them for illegal purposes the user is inheriting their mistakes when caught. - Some of those proxies, pretty all the ones coming from low ranked websites, are tracked so there might be a user privacy violation issue.

2.3.2 Handlers

During the scraping many things could be going wrong. Starting from the things that have been previously explained at the chapter start (URL structure changes, data are moved in different location so that css query goes empty…), ending with the ones that have just been said a few lines ago. Handlers and trycatch error workarounds are explicitly built in this sense. The continuous testing of the scraping functioning while developing has required the maintainer to track where the error occurs. A few numbers: the "agglomerative" function `get.data.catsing()` triggers more than 36 scrapping functions that are going to catch 36 different data pieces. If one of them went missing then the other one would be missing too. Then when row-data is binded together one entry column might not exists making the process fail.

Then the solution to that is to call inside the aggolmerative function as much as trycatch as many scrapping functions are involved. The trycatch can leverage the gap by introducing a specified quantity and alerting that something went wrong. On top of that many other handlers are called throughout the procedure:

- get_ua() verifies that the user agent coming from the session request is not the default one

```
get_ua = function(sess) {
    stopifnot(is.session(sess))
    stopifnot(is_url(sess$url))
    ua = sess$response$request$options$useragent
    return(ua)
}
```

- is_url() verifies that the url input needed has the canonic form. This is done by a REGEX query.

```
is_url = function(url) { re =
    "^(?:(?:http(?:s)?|ftp)://)(?:\\S+(?::(?:\\S)*)?@)?(?:(?:[a-z0-9¡-<ef><U+00BF><U+00BF>](?:-)*)*(?:[a-z0-9¡-<ef><U+00BF><U+00BF>])+)(?:\\.(?:[a-z0-9¡
    grepl(re, url) }
```

- .get_delay() checks through the robotxt file if a delay between each request is kindly welcomed.

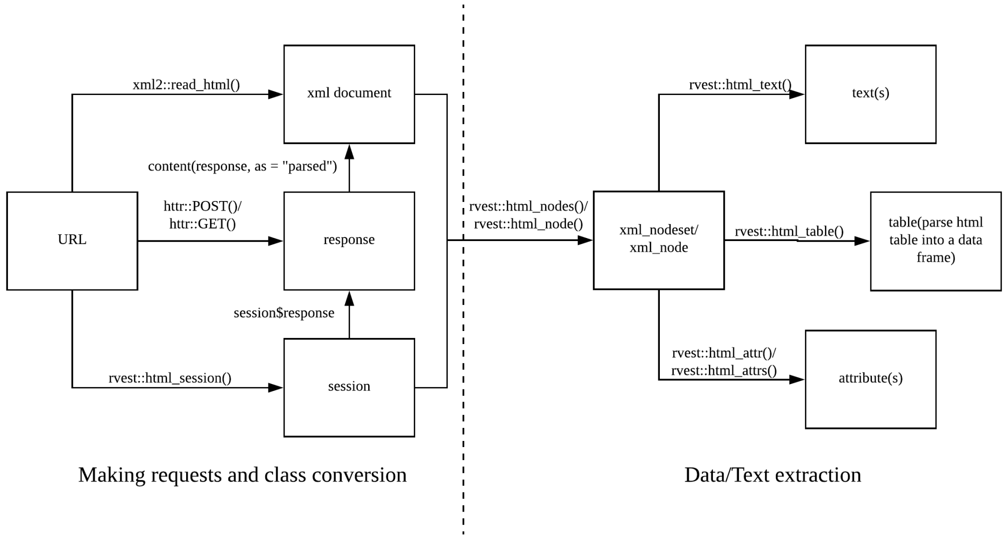
```
.get_delay = function(domain) {

message(sprintf("Refreshing robots.txt data for %s...", domain))

cd_tmp = robotstxt::robotstxt(domain)$crawl_delay

if (length(cd_tmp) > 0) { star = dplyr::filter(cd_tmp,
    useragent=="*") if (nrow(star) == 0) star = cd_tmp[1,]
    as.numeric(star$value[1]) } else { 10L }
}


get_delay = memoise::memoise(.get_delay)
```

2.4 How they are designed with `rvest`

The way scraping functions are designed are walks around three main programming concepts:

- Continuous integration and easy debugging
- Intuitive structure
- Fasten process with respect to the goal of acquiring data

2.4.1 From Generic and Specific structure



Making requests and class conversion

Data/Text extraction

Figure 2.2: functional structure

2.4.2 Parallel Computing

Since many html sessions are opened and within each session many requests are sent computations can take a while. For the sake of the analysis and the app this should not bother the end user because scraping tasks are performed daily and a single day is sufficient amount of runtime. In any case functions are optimized following the criteria stated before. Run time is crucial when dealing with time series and time to market in real estate is very important, this leads to have always fresh data. A way to secure fresh new data is to have lightweight computation on a single machine o heavy computation divided among a bunch of different machines, in this case sessions. A first attempt was using `furrr` package (Vaughan and Dancho, 2018) which enables mapping through a list with the `purrr` , along with a `future` parallel backend. This has shows decent results, but its run time increases when more requests are sent. This leads to a preventive conclusion about the computational complexity: it has to be at least linear. Empirical demonstrations have been made:

```
library(furrr)


vecelaps = c()

start = c()

end = c()

for (i in 1:len(list.of.pages.imm[1:20])) { start[i] = Sys.time()

    cat("\n\n run iteration", i, "over 20 total\n")

    list.of.pages.imm[1:i] %>% furrr::future_map(get.data.caturl,

    .progress = T) %>% bind_rows()


end[i] = Sys.time() vecelaps[i] = end[i] - start[i] }
```

```
furrrmethod = tibble(start,

end,

vettoelaps)
```

```r
# ggplot (themed) run time meausurament method 1
p = ggplot(furrrmethod,aes(x=1:20, y=vettoelaps)) +
geom_line( color="steelblue") +
geom_point() +
xlab("Num URLS evaluated") +
ylab("run time (in seconds)") +
ggtitle("Run-Time for First method (furrr multisession)") +
stat_smooth(method=lm) +
theme_nicco()
p
```

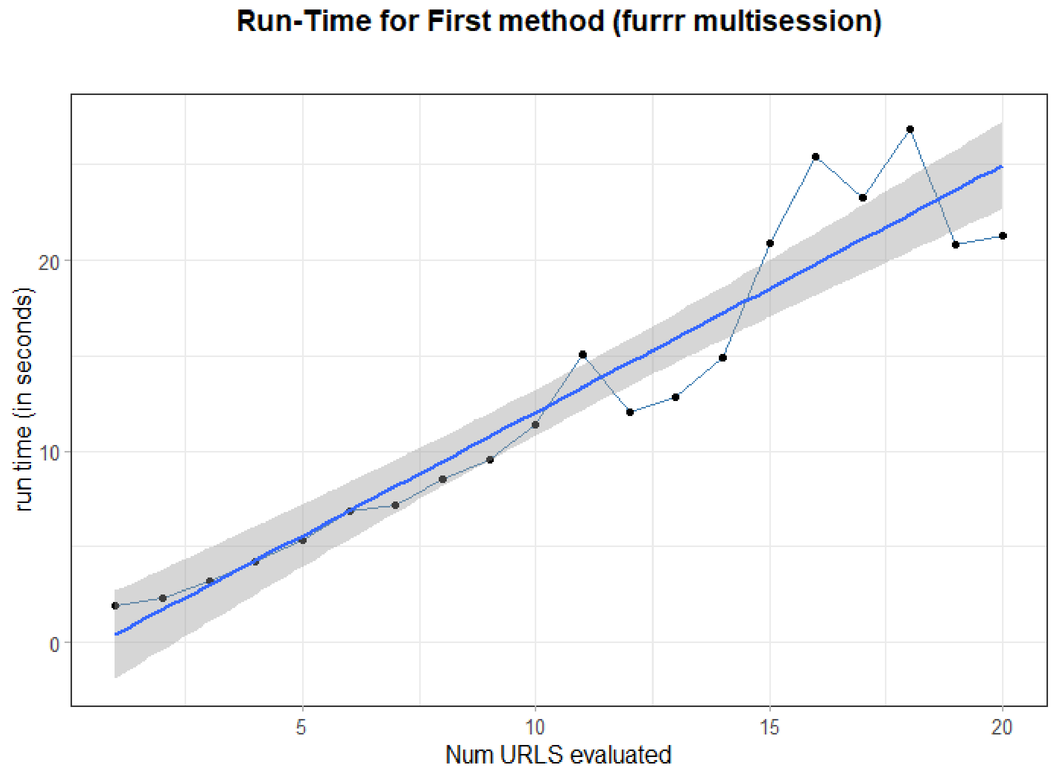


Figure 2.3: computational complexity analysis with Furrr

On the x-axis the number of urls evaluated together, iteration after iteration the urls considered are increased by one. On the y-axis the time measured in seconds. Looking at the smoothing curve in between an easy guess might be linear time $O(n)$.

A second attempt tried to explore the `foreach` package (Microsoft and Weston, 2020). This interesting package enables a new looping construct for executing R code in an iterative way. With all the variety of existing (`apply`, `lapply`, `sapply`, `eapply`, `mapply`, `rapply`,) looping constructs, it might be doubted that there is a need for yet another construct. The main reason for using the `foreach` package is that it supports *parallel execution*, that is, it can execute those repeated operations on multiple processors/cores on the computer, or on multiple nodes of a cluster. The construction is straightforward:

- start clusters on processors cores
- define the iterator, in this case i = to the elements that are going to be iterated through
- `.packages`: Inherits the packages that are used in the tasks define below
- `.combine`: Define the combining function that bind results at the end (say cbind, rbind or tidyverse::bind_rows). It has to be a string.
- `.errorhandling`: specifies how a task evaluation error should be handle.
- `%dopar%`: the dopar keyword suggests foreach with parallelization method
- then the function within the elements are iterated
- close clusters

One major important thing concerns the fact that th function within iterators repeats itself should be standalone. For standalone it is meant that the body function should be defined inside, as it would be a an empty environment. As a matter of fact packages has to be taken inside each time, and if the function is not defined inside body (or is not source from some other locations) the clusters can not operate and an error is printed.

```r
cl = makeCluster(detectCores()-1)
registerDoParallel(cl)


vettoelaps1 = c()
```

```
start1 = c()
end1 = c()
for (j in 1:len(list.of.pages.imm[1:20])) {
start1[j] = Sys.time()
cat("\n\n run iteration",j,"over 20 total\n")
foreach(i = seq_along(list.of.pages.imm[1:j]),
.packages = lista.pacchetti,
.combine = "bind_rows",
.errorhandling='pass') %dopar% {
source("main.R")
x = get.data.caturl(list.of.pages.imm[i])
}
end1[j] = Sys.time()
vettoelaps1[j] = end1[j] -start1[j]
}
stopCluster(cl)
```

It can be seen quite easily that the curve is flattened and resembles someway logarithmic time $O(log(n))$.

A further improvement could be obtained using a new package called `doAzureParallel` which is built on top of foreach. doAzureParallel enables different Virtual Machines operates parallel computing throughout Microsoft Azure cloud, but this comes at a substantial monetary cost. This would be a perfect match given that parallel methods seen before accelerates the number of requests sent among different processors or cluster, even though actually what it is really needed it is something that separates session. Unleashing Virtual Machines permits from one hand to further increase computational power and the number of potential requests, from the other it can splits requests among different user agents (a pool for each VM) masquerading even better the scraping automation.

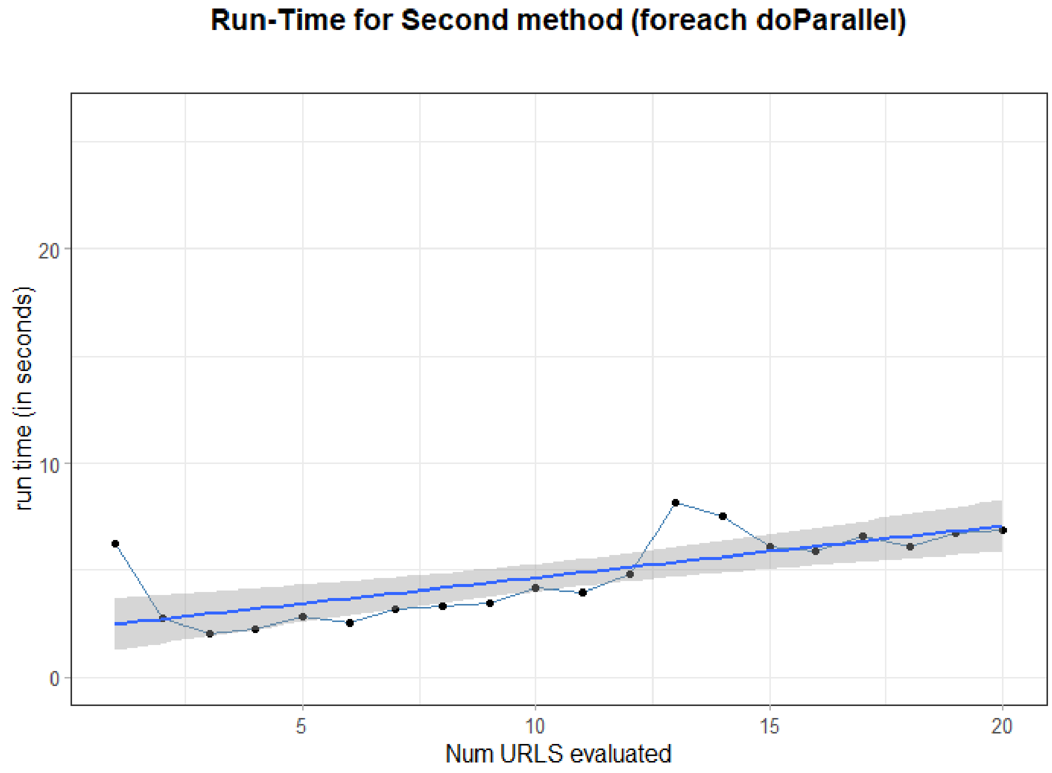


Figure 2.4: computational complexity analysis with Furrr

2.5 What are the Advantages of this Workflow

2.6 Legal Challenges (ancora non validato)

“Data that are online and public are always available for all” is never a good answer to the question “Can I use that data to my scope”. Immobiliare.it[1] is not providing any source of data from its own database neither it is planning to do so in the future. It has not even provided a paid API through which might be possible to perform analysis. However the golden standard for scraping was respected since the robot.txt file is clear allowing any actions as demonstrated above. What it worth noting is that some other popular player other like Idealista is using a different approach. Some of procedures that has been applied to the immboliare scraping was not possibile on the Idealista. This could be due to many reason:

[1] https://www.immobiliare.it/

- Idealista content is composed by Javascript so and html parser can no get that.
- Idealista blocks also certain web browser that have a demonstrated “career” in scraping procedures.

All of this leads to accept that entry barriers to scrape are for sure higher than the one faced for Immobiliare. The reticence to share data could be a reflex on how big idealista is; as a matter of fact it has a heavy market presence in some of the Europe real estates country as Spain and France. So what they thought was to raise awareness on scraping procedure that in a certain way can hurt their business. This has been validated by the fact that prior filtering houses on their website a checkbox has to be signed. The checkbox make the user sign an agreement on their platform according to which data can not be misused and it belongs their intellectual property.

Chapter 3

Infrastructure

In order to provide a fast, portable and integrated product to the end user It has been designed a quite straightforward software architecture. We have already seen the scraping functions and how they are built around the concept of easy flexibility and debugging. This is due to the fact that they should extract something that is dynamic, it is not sure that it will be as the day before. The data we are trying to grab might have been moved somewhere else throughout the website. Or it might have placed extra expression inside the node we are inspecting ("$" sign following the monthly rental price) . A very often occurring example regards the way information concerning the house are represented in the website. Considering the september 2018 january 2020 time span the design of the website has changed a vast number of times. Since both the design and the scrapping functions relies on the HTML skelethon and CSS queries. As soon as something changes in the website the other files needs to be readjusted to be consistent with the content and so back and forth. The debugging handlers nested in the functions helps the maintainer to grasp what it is not working properly where the error occur. The same inner philosophy has been applied to the software architecture chosen for this project. First of all the wide range of open source solutions (back-end and front-end) and documentation on this has made many analyst and data scientist almost full stack developer. This was also due to the fact that RStudio has set very well oriented guidelines spending a lot of effort giving its users an easy, integrated and interconnected environment. By that it is meant that recently the RStudio community has developed, on top of many different others, an entire package dedicated to REST APIs (Plumber (Trestle Technology, LLC, 2018)). MOreover developers in RStudio and its contributors have created an entire new paradigm called Shiny (Chang et al., 2020), a popular web app development package, that forces the user to have front-end and back-end technologies tied up in the same IDE (RStudio) and with a unique language to deal with. The front end file (for simplicity named UI.R) contains the UI's layout and the style and also other javascripts components. On the other hand the server file (named after server.R) absorbs the back-end, under the hood, code and makes the UI intercact and respond to the user. This comes at a cost of flexibility and customization since Shiny could not easily handle too many embellishments (even though potentially can). Nevertheless a unique environment makes integration with other technologies easier and most of all introduces the analyst to a full stack approach. Many open source projects are gravitating around the Shiny framework with the aim to extend its capabilities. One example is a newly created package called reactR (Inc et al., 2020) that allows user to implement the power of React.js into the shiny UI front end. All of this is possible, once again, by the R community but a greater contribution come from digging up the right path along which everything by open source comes natural. (parallelo con la vigna e l'albero che la sostiene e indirizza) The carrier idea for this project is to have parallelized scraping functions called daily by a scheduler producing and subsequently storing a .csv file in a MySQL /cartoDB database. They are all tought to be containerized in a Linux (Ubuntu distr) docker container hosted in a AWS EC2 server. Then in a second container a Shiny app is placed, this one pipes in data from the former infrastructure and apply the statistical model stored (by an API call) in its server.R part.

The main technologies implied are:

- Scheduler cron job

- Docker containers
- Shiny
- Plumber REST API
- AWS (Amazon Web Services) EC2
- CartoDB

On top of that even each single part of this thesis has been made stand alone and can be easily accessed and modified through this link[1]. The pdf (theis) version of the gitbook can be obtained by clicking the download button that can be seen in figure below. A Latex engine (Xelatex) wrapped into the website compiles a sequence of Markdown documents converting them into .html (the book's chapters) which are formatted by rules grouped in a .yml file. All the documents are pushed to a Github repository with git. By a simple trick, since all the files are static html, they can be displayed through GH pages as it is a website. All of this has been possible thanks to Bookdown (Xie, 2020) once again a R well documented package (Xie, 2016) to build interactive books along with RMarkdown (Allaire et al., 2020).

An empirical observation of immobiliare.it has suggested that houses rents advertisement are continuously added and then removed during the day. Fresh data is needed to have updated analysis since the scope in here is to offer realtime considerations. Something should be automated periodically in order to address the issue. Moreover, as rule of thumb, a daily data extraction might be a good option for some reasons. It can intercept price variations with a relatively small time lag, It can also display some sort of pattern in time that would help the reader/user to select the perfect choice. As a consequence a daily .csv file is generated and directly collected into a Db folder arranged by time The solutiond proposed takes care of the issue by making the scraping script generating the .csv be executed by a scheduler.

[1] https://niccolosalvini.github.io/Thesis/

3.1 Scheduler

A Scheduler in a process is a component on a OS that allows the computer to decide which activity is going to be executed. In the context of multi-programming it is thought as a tool to keep CPU occupied as much as possible. As an example it can trigger a process while some other is still waiting to finish. There are many type of scheduler and they are based on the frequency of times they are executed considering a certain closed time neighbor.

- Short term scheduler: it can trigger and queue the "ready to go" tasks
  - with pre-emption
  - without pre-emption

The ST scheduler selects the process and It gains control of the CPU by the dispatcher. OIn this context we can define latency as the time needed to stop a process and to start a new one.

- Medium term scheduler
- Long term scheduler

for some other useful but beyond the scope information, such as the scheduling algorithm the reader can refer to (Wikiversità, 2020).

The scheduler in this context cosists in a .sh (shell file, sort of text file) composed by a set of instructions that are being executed by the computer on daily basis. This file has to be in the same WD (working directory) of the project in order to make it working. Some common issues can occur when new files coming after the execution of the scheduled main script are generated, but the path isnt explicitly specified. This can lead to the partial or incomplete generation of the file since the shell file is executed within the folder but is triggered by some other location on the computer. Each OS has its own scheduler and syntax to call it. Since we are interested in Ubuntu machines the scheduler is said to be a cron job. Later it will be clear why Ubuntu is the option to pursue.

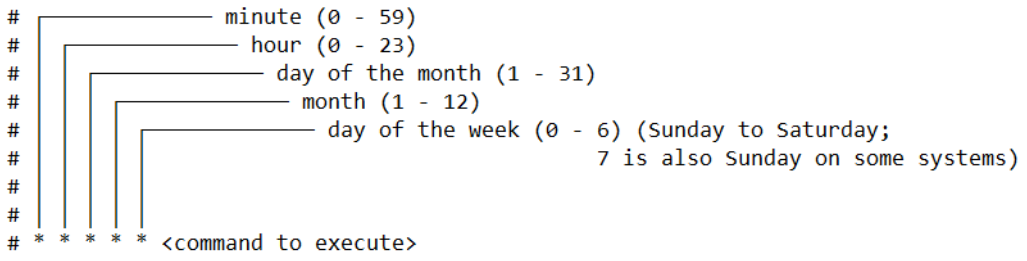
va parafrasato

3.1.1 Cron Jobs

The software utility cron also known as cron job is a time-based job scheduler in Unix-like computer operating systems. Users that set up and maintain software environments use cron to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It typically automates system maintenance or administration—though its general-purpose nature makes it useful for things like downloading files from the Internet and downloading email at regular intervals.

The actions of cron are driven by a crontab (cron table) file, a configuration file that specifies shell commands to run periodically on a given schedule. The crontab files are stored where the lists of jobs and other instructions to the cron daemon are kept. Users can have their own individual crontab files and often there is a system-wide crontab file (usually in /etc or a subdirectory of /etc) that only system administrators can edit.

Each line of a crontab file represents a job, and looks like this:



```
# ┌───────────── minute (0 - 59)
# │ ┌───────────── hour (0 - 23)
# │ │ ┌───────────── day of the month (1 - 31)
# │ │ │ ┌───────────── month (1 - 12)
# │ │ │ │ ┌───────────── day of the week (0 - 6) (Sunday to Saturday;
# │ │ │ │ │                                   7 is also Sunday on some systems)
# │ │ │ │ │
# │ │ │ │ │
# * * * * * <command to execute>
```

Figure 3.1: crontab

Each line of a crontab file represents a job. This example runs a shell program called scheduler.sh at 23:45 (11:45 PM) every Saturday.

45 23 * * 6 /home/oracle/scripts/scheduler.sh

Some rather unusual scheduling definitions and syntax for cronjobs can be found in this reference (Wikipedia contributors, 2020)

The cron job applied to the script needs to be ran at 11:30 PM everyday. It has that forms: —> qui immagine

va parafrasato

For now the computational power comes from the machine on which the system is installed. A smarter solution takes into consideration that the former infrastructure has its own limits. Major limits comprehend run time since at the same moment the machine runs locally both the scraping functions and the app computations. This to a certain extent might fit for personal use but as data increases all the system risks to fail. It is also totally local so the analysis can not be shared with anyone. This problem can be addressed with a technology that has seen a huge growth in its usage in the last few years: Docker containers.

3.2 Docker Container

from docker In 2013, Docker introduced what would become the industry standard for containers. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

3.2.1 What is Docker?

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

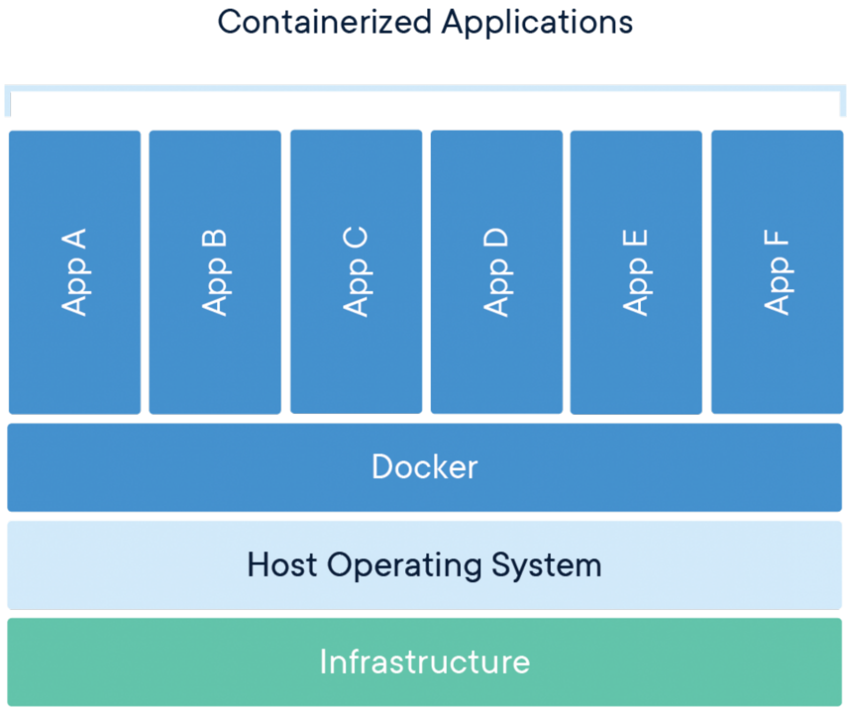


Figure 3.2: docker example

Docker leveraged existing computing concepts around containers and specifically in the Linux world. Docker's technology is unique because it focuses on the requirements of developers and systems operators to separate application dependencies from infrastructure.

A question might come up about why a Virtual Machine could not be a preferable container for our specified task. Well, Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient.

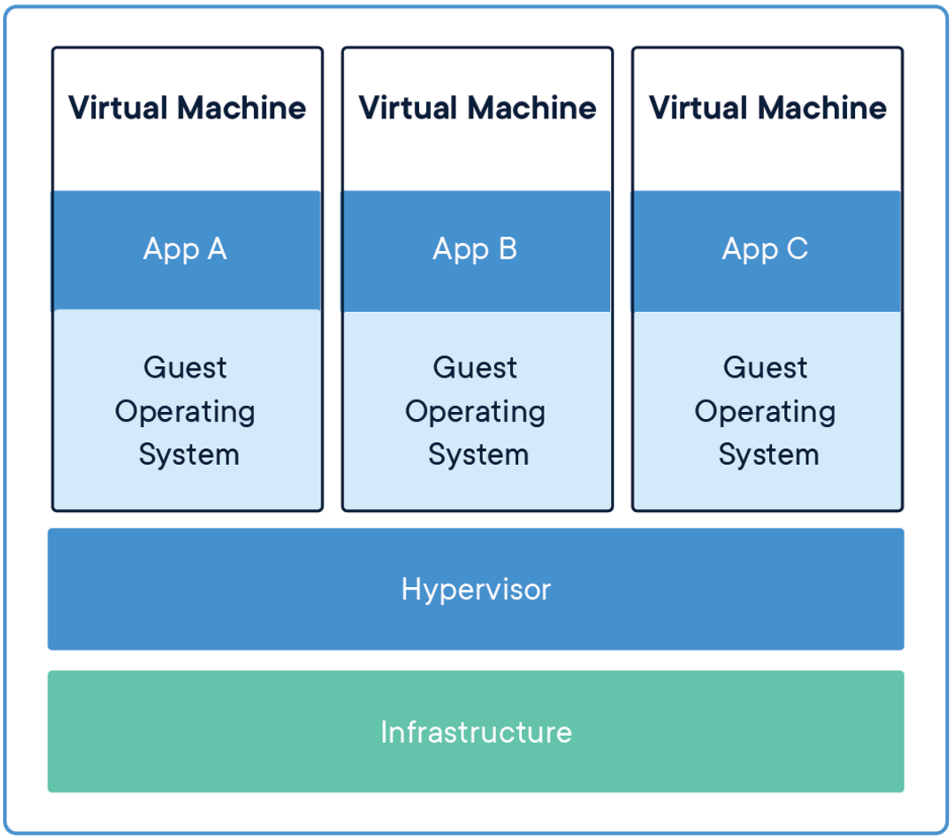


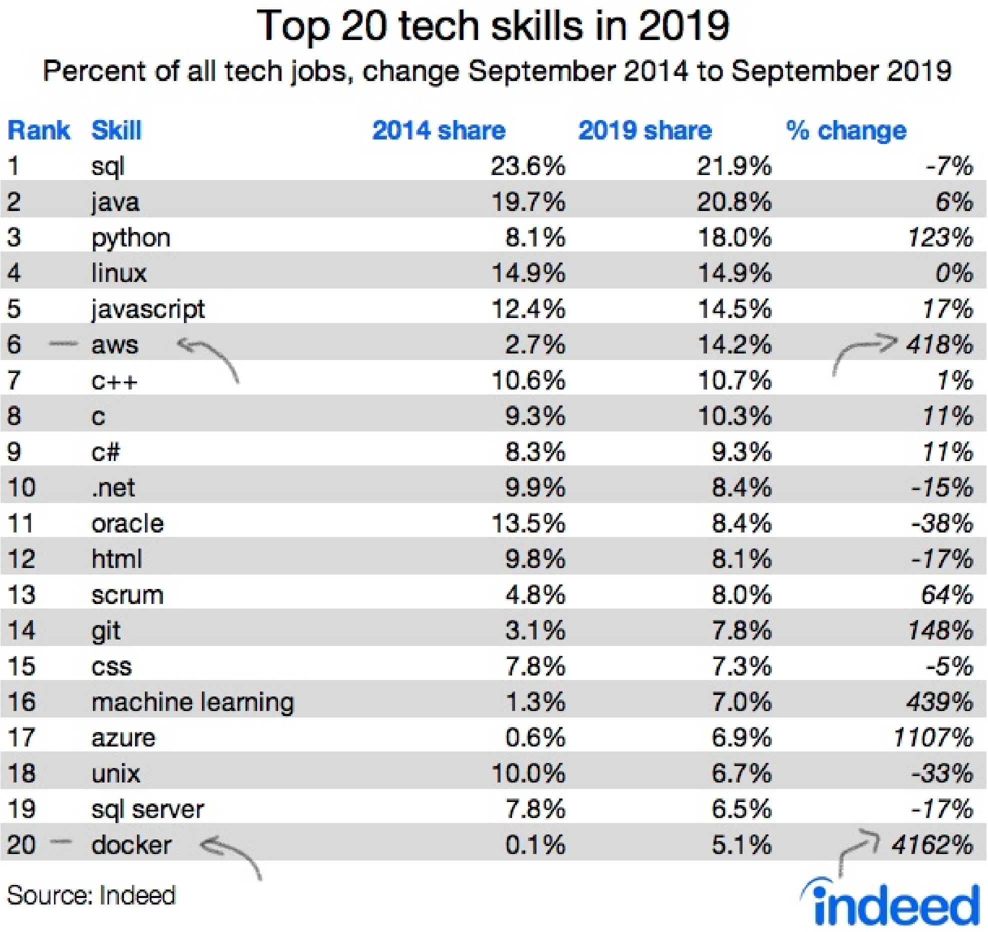
Figure 3.3: docker container vs VM

from docker

3.2.2 What are the main andvantages of using Docker

va parafrasato from Matt Dancho

Indeed, the popular employment-related search engine, released an article this past Tuesday showing changing trends from 2015 to 2019 in "Technology-Related Job Postings". We can see a number of changes in key technologies - One that we are particularly interested in is the 4000% increase in Docker.



Top 20 tech skills in 2019

Percent of all tech jobs, change September 2014 to September 2019

Rank	Skill	2014 share	2019 share	% change
1	sql	23.6%	21.9%	-7%
2	java	19.7%	20.8%	6%
3	python	8.1%	18.0%	123%
4	linux	14.9%	14.9%	0%
5	javascript	12.4%	14.5%	17%
6	aws	2.7%	14.2%	418%
7	c++	10.6%	10.7%	1%
8	c	9.3%	10.3%	11%
9	c#	8.3%	9.3%	11%
10	.net	9.9%	8.4%	-15%
11	oracle	13.5%	8.4%	-38%
12	html	9.8%	8.1%	-17%
13	scrum	4.8%	8.0%	64%
14	git	3.1%	7.8%	148%
15	css	7.8%	7.3%	-5%
16	machine learning	1.3%	7.0%	439%
17	azure	0.6%	6.9%	1107%
18	unix	10.0%	6.7%	-33%
19	sql server	7.8%	6.5%	-17%
20	docker	0.1%	5.1%	4162%

Source: Indeed

indeed

Figure 3.4: docker-stats

The landscape of Data Science is changing (was previously an Economist at the Indeed Hiring Lab, 2020) from reporting to application building:

In 2015 - Businesses need reports to make better decisions In 2020 - Businesses need apps to empower better decision making at all levels of the organization This transition is challenging the Data Scientist to learn new technologies to stay relevant…

As a matter of fact, it is no longer sufficient to just know machine learning algorithms. Future data workers need to know how to put machine learning into production as quickly as possible to meet the business needs. This can be done either integrating existing obsolete/old technologies with the new ones, or build a solid, portable and scalable infrastructure to better the processes. In order to do so, It is strictly needed to learn from programmers the basics of Software Engineering. The extremely good news is that no data scientist whatsoever needs to be a perfect software engineer, but at least he has to know how to integrate already built techonlogies with its work. This truly can help in the quest to unleash data science at scale and unlock real business value. The way Docker does that are many (red, 2020):

- *Rapid application deployment* : containers include the minimal run time requirements of the application, reducing their size and allowing them to be deployed quickly.
- *Portability across machines* :an application and all its dependencies can be bundled into a single container that is independent from the host version of Linux kernel, platform distribution, or deployment model. This container can be transfered to another machine that runs Docker, and executed there without compatibility issues.
- *Version control and component reuse* : you can track successive versions of a container, inspect differences, or roll-back to previous versions. Containers reuse components from the preceding layers, which makes them noticeably lightweight.
- *Sharing* : you can use a remote repository to share your container with others. It is also possible to configure a private repository hosted on Docker Hub.

- *Lightweight footprint and minimal overhead* : Docker images are typically very small, which facilitates rapid delivery and reduces the time to deploy new application containers.
- *Simplified maintenance* :Docker reduces effort and risk of problems with application dependencies.

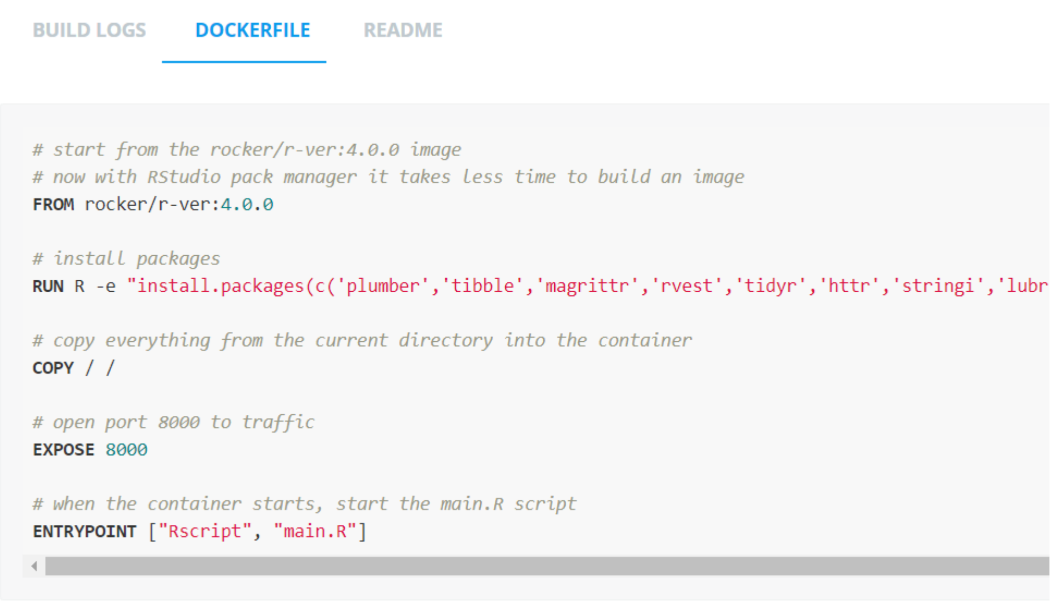
The way all of this is possible is a dockerfile that determines the instruction that docker has to perform to abstract the environment.

3.2.3 Dockerfile

Docker can build images automatically by interpreting the instructions from a Dockerfile. A Dockerfile can be thought as a written recipe to cook a specific cake, with all the ingredients described in a piece of a paper using a generic oven and adding layer of preparation after layer. A Dockerfile is a text format document that contains all the commands/rules a generic user could call on the CLI to assemble an image. Executing the command `docker build` from shell the user can trigger the image building. That executes several command-line instructions in chronological succession of steps. The Dockerfile used to trigger the build of the docker image has this following set of instructions:

- `FROM rocker/r-ver:4.0.0` : the command imports an image already written by the rocker team (authored contributors for the R docker project) that contains the base-R version 4.0.0. Recently with the 4.0 version the RStudio team has created a repository management server for its packages that organizes and centralizes R packages (offline access and checkpoints). This will shorten the installation time and secure packages since they all can be freezed into a version that make the whole system works.

- `RUN R -e "install.packages(c('plumber','tibble','...',dependencies=TRUE)` : the command install all the packages required to execute the files (R



```
# start from the rocker/r-ver:4.0.0 image
# now with RStudio pack manager it takes less time to build an image
FROM rocker/r-ver:4.0.0

# install packages
RUN R -e "install.packages(c('plumber','tibble','magrittr','rvest','tidyr','httr','stringi','lubr

# copy everything from the current directory into the container
COPY / /

# open port 8000 to traffic
EXPOSE 8000

# when the container starts, start the main.R script
ENTRYPOINT ["Rscript", "main.R"]
```

Figure 3.5: dockerfile

files) containerized for the scraping. Since all the packages have their dependencies the option `dependencies=TRUE` is needed.

- `EXPOSE 8000` : the commands instructs Docker that the container listens on the specified network ports 8000 at runtime. It is possible to specify whether the port exposed listens on UDP or TCP, the default is TCP (this part needs a previous set up previous installing, for further online documentation It is reccomended (doc, 2020))

- `ENTRYPOINT ["Rscript", "main.R"]` : the command tells docker to execute the Rscript extension file main.R within the container that triggers the API building/the generation of the .csv file.

va parafrasato from Matt Dancho

An alternative and very used approach could be wrapping all the scraping function into an API and then send a `GET` request to the API endpoint needed.

3.3 API

va parafrasato

The scraping functions, according to how the author as structured them, are able to produce two .csv extension (if the boolean option `write` is set = TRUE) files. As already clarified in the previous ssection some point should be joined by a primary key. But for the sake of In order to give the possibility to have a daily updated saptial analysis on data we need to continously have fresh data. In the website data come and go, as products in a marketplace, so the main idea is to have something that catches the new added and deletes what it is already taken. Nowadays we have many open source, nearly cost free, techonlogies that allow us to have corporate grade applications that can be orizontally scaled at need. Most of them come with great docuemntation and ready to use examples that flatten the learning curve. The first choice that has to be made is: either to provide a .csv file day by day with all the data to feed the application, or we exploit some portable and fast solutions as API.

va parafrasato

3.4 What is an API

API is a set of definitions and protocols for building and integrating application software. API stands for application programming interface.

APIs let a product or a service communicate with other products and services without having to know how they're implemented. This can simplify app development, saving time and money. When you're designing new tools and products—or managing existing ones—APIs give flexibility; simplify design, administration, and use; and provide opportunities for innovation. APIs are sometimes thought of as contracts, with documentation that represents an agreement between parties: If party 1 sends a remote request structured a particular way, this is how party 2's software will respond.

API examples: - Google Maps API: allows developers embed geo-location data using JavaScript. The Google Maps API is designed to work on mobile and desktop. - YouTube API: allows developers integrate YouTube videos and functionalities into websites or applications. - Google Analytics API: allows to track website performance in terms of audience, monetization and other important metrics throught the Google Analytics interface. The website the thesis come from has this implementation working.

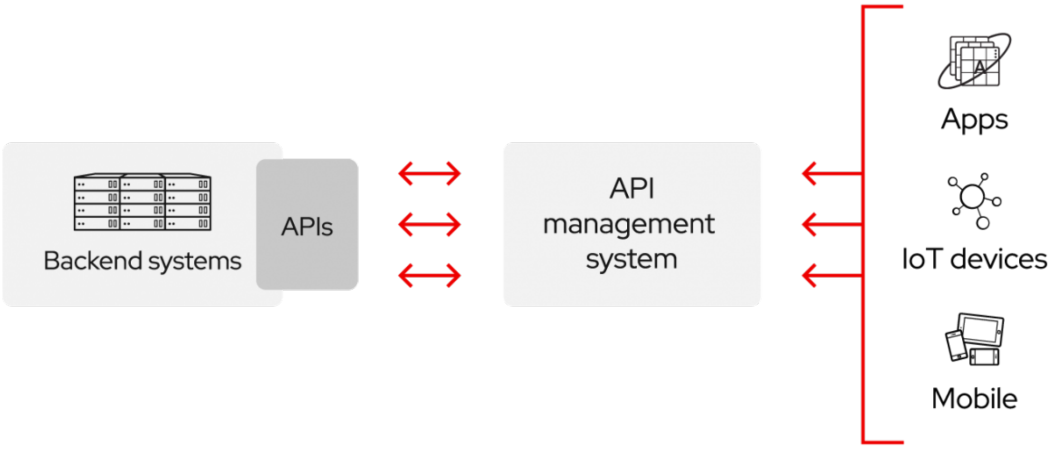


Figure 3.6: API functioning

Because APIs simplify how developers integrate new application components into an existing architecture, they help business and IT teams collaborate. Business needs often change quickly in response to ever shifting digital markets, where new competitors can change a whole industry with a new app. In order to stay competitive, it’s important to support the rapid development and deployment of innovative services. Cloud-native application development is an identifiable way to increase development speed, and it relies on connecting a microservices application architecture through APIs.

3.4.1 What in practice an API does

API, strictly talking to this thesis extent, are cloud infrastructure that given a specific URL (and the most of the times credentials) are outputting data. Data comes in a given format (the most of the times JSON),so that they can be instantly pre-processed and elaborated, in this case by oine other software, the Shiny app.

(va prafrasato)

Since website are continuously changed for many reasons API philosophy results in a smarter choice since it makes easy to access data and flex API endpoints to the business needs.

3.4.2 Plumber API

Plumber (api, 2020) allows the user to create a web API by simply adding decoration comments to the existing R code. Decorations are a special type of comments that suggests the plumber where and when the API parts are. Here below a toy example from the reference:

```
# plumber.R file

# * Echo back the input * @param msg The message to echo * @get /echo
function(msg = "") {
    list(msg = paste0("The message is: '", msg, "'"))
}

# * Plot a histogram * @png * @get /plot
function() {
    rand = rnorm(100)
    hist(rand)
}

# * Return the sum of two numbers * @param a The first number to add * @param
# The second number to add * @post /sum
function(a, b) {
    as.numeric(a) + as.numeric(b)
}
```

This chunk of code assembled into a .R file has three endpoints where interruption occurs.

Special comments are marked as this `#*` and they are followed by specific keywords denoted with `@`. - the name of the API (unique without the `@`) - the `@params` keyword refers to parameter that has to be inputted to give the result of the function define below. If in the function below defualt parameters are stated then the API response is the elaboration of the functions with those parameters. If the function does not specify any parameter, see the below endpoints below, plumber has to make sure that the function can run without them. - the `#* @png` chuck piece specify the extension of the output file. - the `#* @get /plot` decorations specify the type of HTTP request we are sending, in this case a GET request. The user in this case is requesting some information to the API, the name is plot, so the expectations are that a plot is given as response

Chapter 4

Spatial Statistics

4.1 Gentle Introduction

va parafrasato Researchers in diverse areas such as climatology, ecology, environmental health, and real estate marketing are increasingly faced with the task of analyzing data that are: - highly multivariate, with many important predictors and response variables, - geographically referenced, and often presented as maps, and - temporally correlated, as in longitudinal or other time series structures. For example, for an epidemiological investigation, we might wish to analyze lung, breast, colorectal, and cervical cancer rates by county and year in a particular state, with smoking, mammography, and other important screening and staging information also available at some level. Public health professionals who collect such data are charged not only with surveillance, but also statistical inference tasks, such as modeling of trends and correlation structures, estimation of underlying model parameters, hypothesis testing (or comparison of competing models), and prediction of observations at unobserved times or locations. In this text we seek to present a practical, self-contained treatment of hierarchical mod- eling and data analysis for complex spatial (and spatiotemporal) datasets. Spatial statistics methods have been around for some time, with the landmark work by Cressie (1993) pro- viding arguably the only comprehensive book in the area. However, recent developments in Markov chain Monte Carlo (MCMC) computing now allow fully Bayesian analyses of sophisticated multilevel models for complex geographically referenced data. This approach also offers full inference for non-Gaussian spatial data, multivariate spatial data, spatiotem- poral data, and, for the first time, solutions to problems such as geographic and temporal misalignment of spatial data layers. **va parafrasato**

4.2 INLA estimation

va parafrasato For many years, Bayesian inference has relied upon Markov chain Monte Carlo methods (Gilks et al. 1996; Brooks et al. 2011) to compute the joint posterior distribution of the model parameters. This is usually computationally very expensive as this distribution is often in a space of high dimension.

Havard Rue, Martino, and Chopin (2009) propose a novel approach that makes Bayesian inference faster. First of all, rather than aiming at estimating the joint posterior distribution of the model parameters, they suggest focusing on individual posterior marginals of the model parameters. In many cases, marginal inference is enough to make inference of the model parameters and latent effects, and there is no need to deal with multivariate posterior distributions that are difficult to obtain. Secondly, they focus on models that can be expressed as latent Gaussian Markov random fields (GMRF). This provides the computational advantages (see Rue and Held 2005) that reduce computation time of model fitting. Furthermore, Havard Rue, Martino, and Chopin (2009) develop a new approximation to the posterior marginal distributions of the model parameters based on the Laplace approximation (see, for example, MacKay 2003). A recent review on INLA can be found in Rue et al. (2017) **va parafrasato**

4.3 Presentation of data

4.4 Point-referenced data models

Chapter 5

Point Referenced Data

5.1 Point-Referenced modeling

5.1.1 Stationarity

5.1.2 Variograms

5.1.3 Isotropy

5.1.4 Variogram model fitting

5.2 Anisotropy

5.3 Exploratory analysis

Chapter 6

Bayesian Spatial Modelling

va parafrasatoo

Several types of models are used with spatial and spatio-temporal data, depend- ing on the aim of the study. If we are interested in summarizing spatial and spatio-temporal variation between areas using risks or probabilities then we could use statistical methods like disease mapping to compare maps and identify clusters. Moran Index is extensively used to check for spatial autocorrelation (Moran, 1950), while the scan statistics, implemented in SaTScan (Killdorf, 1997), has been used for cluster detection and to perform geographical surveillance in a non-Bayesian approach. The same types of models can also be used in studies where there is an aetiological aim to assess the potential effect of risk factors on outcomes. A different type of study considers the quantification of the risk of experienc- ing an outcome as the distance from a certain source increases. This is typically framed in an environmental context, so that the source could be a point (e.g., waste site, radio transmitter) or a line (e.g., power line, road). In this case, the meth- ods typically used vary from nonparametric tests proposed by Stone (1988) to the parametric approach introduced by Diggle et al. (1998). In a different context, when the interest lies in mapping continuous spatial (or spatio-temporal) variables, which are measured only at a finite set of specific points in a given region, and in predicting their values at unobserved locations, geostatistical methods – such as kriging – are employed (Cressie, 1991; Stein, 1991). This may play a significant role in environmental risk assessment in order to identify areas where the risk of exceeding potentially harmful thresholds is higher. Bayesian methods to deal with spatial and spatio-temporal data started to appear around year 2000, with the development of Markov chain Monte Carlo (MCMC) simulative methods (Casella and George, 1992; Gilks et al., 1996). Before that the Bayesian approach was almost only used for theoretical models and found little applications in real case studies due to the lack of numerical/analytical or simula- tive tools to compute posterior distributions. The advent of MCMC has triggered the possibility for researchers to develop complex models on large datasets without INTRODUCTION 3 the need of imposing simplified structures. Probably the main contribution to spatial and spatio-temporal statistics is the one of Besag et al. (1991), who developed the Besag–York–Mollié (BYM) method (see Chapter 6) which is commonly used for disease mapping, while Banerjee et al. (2004), Diggle and Ribeiro (2007) and Cressie and Wikle (2011) have concentrated on Bayesian geostatistical models. The main advantage of the Bayesian approach resides in its taking into account uncertainty in the estimates/predictions, and its flexibility and capability of dealing with issues like missing data. In the book, we follow this paradigm and introduce the Bayesian philosophy and inference in Chapter 3, while in Chapter 4 we review Bayesian computation tools, but the reader could also find interesting the follow- ing: Knorr-Held (2000) and Best et al. (2005) for disease mapping and Diggle et al. (1998) for a modeling approach for continuous spatial data and for prediction

va parafrasatoo

6.1 INLA

For many years, Bayesian inference has relied upon Markov chain Monte Carlo methods (Gilks et al. 1996; Brooks et al. 2011) to compute the joint posterior distribution of the model parameters. This is usually computationally very expensive as this distribution is often in a space of high dimension.

Havard Rue, Martino, and Chopin (2009) propose a novel approach that makes Bayesian inference faster. First of all, rather than aiming at estimating the joint posterior distribution of the model parameters, they suggest focusing on individual posterior marginals of the model parameters. In many cases, marginal inference is enough to make inference of the model parameters and latent effects, and there is no need to deal with multivariate posterior distributions that are difficult to obtain. Secondly, they focus on models that can be expressed as latent Gaussian Markov random fields (GMRF). This provides the computational advantages (see Rue and Held 2005) that reduce computation time of model fitting. Furthermore, Havard Rue, Martino, and Chopin (2009) develop a new approximation to the posterior marginal distributions of the model parameters based on the Laplace approximation (see, for example, MacKay 2003). A recent review on INLA can be found in Rue et al. (2017).

6.2 Laplace Approximation

An alternative approach to the simulation-based MC integration is analytic approx- imation with the Laplace method. Suppose we are interested in computing the following integral:

$$\int f(x)\mathrm{d}x = \int \exp(\log f(x))\mathrm{d}x$$

where $f(x)$ is the density function of a random variable X. We represent $logf(x)$ by means of a Taylor series expansion evaluated in x = x0:

$$\log f(x) \approx \log f\left(x_0\right) + \left(x - x_0\right) \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x_0} + \frac{\left(x - x_0\right)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x_0}$$

If x0 is set equal to the mode $x* = argmax$, log f(x) then log f(x) $\frac{\partial \log f(x)}{\partial x}\Big|_{x=x^*} = 0$ and the approximation becomes

$$\log f(x) \approx \log f\left(x^*\right) + \frac{\left(x-x^*\right)^2}{2} \frac{\partial^2 \log f(x)}{\partial x^2}\Bigg|_{x=x^*}$$

The integral of interest is then approximated as follows:

$$\int f(x)\mathrm{d}x \approx \int \exp\left(\log f\left(x^*\right) + \frac{\left(x-x^*\right)^2}{2} \frac{\partial^2 \log f(x)}{\partial x^2}\Bigg|_{x=x^*}\right)\mathrm{d}x$$

$$= \exp\left(\log f\left(x^*\right)\right) \int \exp\left(\frac{\left(x-x^*\right)^2}{2} \frac{\partial^2 \log f(x)}{\partial x^2}\Bigg|_{x=x^*}\right)\mathrm{d}x$$

where the integrand can be associated with the density of a Normal distribution. In fact, by setting

$$\sigma^{2*} = -1/\left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*}$$

we obtain:

$$\int f(x)\mathrm{d}x \approx \exp\left(\log f\left(x^*\right)\right) \int \exp\left(-\frac{\left(x-x^*\right)^2}{2\sigma^{2*}}\right)\mathrm{d}x$$

where the integrand is the kernel of a Normal distribution with mean equal to x∗ and variance σ^{2*}. More precisely, the integral evaluated in the interval (α, β) is approximated by:

$$\int_\alpha^\beta f(x)\mathrm{d}x \approx f\left(x^*\right)\sqrt{2\pi\sigma^{2*}}(\Phi(\beta) - \Phi(\alpha))$$

where $\Phi(\cdot)$ denotes the cumulative density function of th $Normal(x_i, \sigma^{2*})$ distri- bution.

qui volendo un esempio fatto da me

6.3 The Class of Latent Gaussian Models

The first step in defining a latent Gaussian model within the Bayesian framework is to identify a distribution for the observed data y = (y1, ... , yn). A very general approach consists in specifying a distribution for yi characterized by a parameter i (usually the mean E(yi)) defined as a function of a structured additive predictor i through a link function g( ), such that g(i) = i. The additive linear predictor i is defined as follows:

$$\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l \left(z_{li}\right)$$

Here 0 is a scalar representing the intercept; the coefficients $\beta = \{\beta_1, \dots, \beta_M\}$ quantify the (linear) effect of some covariates $x = (x_1, \dots, x_M)$ on the response; and $f = \{f_1(\cdot), \dots, f_L(\cdot)\}$ is a collection of functions defined in terms of a set of covariates $z = (z_1, ..., z_L)$. The terms $f_l(\cdot)$ can assume different forms such as smooth and

nonlinear effects of covariates, time trends and seasonal effects, random intercept and slopes as well as temporal or spatial random effects. For this reason, the class of latent Gaussian models is very flexible and can accomodate a wide range of models ranging from generalized and dynamic linear models to spatial and spatio-temporal models (see Martins et al., 2013 for a review). We collect all the latent (nonobservable) components of interest for the inference in a set of parameters named defined as $\theta = \{\beta_0, \beta, f\}$. Moreover, we denote with $\psi = \{\psi_1, \dots, \psi_K\}$ the vector of the K hyperparameters. By assuming conditional independence, the distribution of the n observations (all coming from the same distribution family) is given by the likelihood

$$p(y \mid \theta, \psi) = \prod_{i=1}^{n} p\left(y_i \mid \theta_i, \psi\right)$$

where each data point yi is connected to only one element i in the latent field θ. Martins et al. (2013) discuss the possibility of relaxing this assumption assuming that each observation may be connected with a linear combination of elements in θ; moreover, they take into account the case when the data belong to several distri- butions, i.e., the multiple likelihoods case.

We assume a multivariate Normal prior on with mean and precision matrix $Q(\psi)$, i.e., $\theta \sim \text{Normal}\left(\mathbf{0}, Q^{-1}(\psi)\right)$ with density function given by

$$p(\theta \mid \psi) = (2\pi)^{-n/2} |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta' Q(\psi)\theta\right)$$

where | | denotes the matrix determinant and is used for the transpose operation. The components of the latent Gaussian field are supposed to be conditionally independent with the consequence that $Q(\psi)$ is a sparse precision matrix.8 This specification is known as Gaussian Markov random field (GMRF, Rue and Held, 2005). Note that the sparsity of the precision matrix gives rise to computational ben- efits when making inference with GMRFs. In fact, linear algebra operations can be performed using numerical methods for sparse matrices, resulting in a considerable computational gain (see Rue and Held, 2005 for algorithms). The joint posterior distribution of and ψ is given by the product of the likelihood (4.13), of the GMRF density (4.14) and of the hyperparameter prior distribution $p(\psi)$:

$$\begin{aligned}
p(\theta, \psi \mid y) &\propto p(\psi) \times p(\theta \mid \psi) \times p(y \mid \theta, \psi) \\
&\propto p(\psi) \times p(\theta \mid \psi) \times \prod_{i=1}^{n} p\left(y_i \mid \theta_i, \psi\right) \\
&\propto p(\psi) \times |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta' Q(\psi)\theta\right) \times \prod_{i=1}^{n} \exp\left(\log\left(p\left(y_i \mid \theta_i, \psi\right)\right)\right) \\
&\propto p(\psi) \times |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta' Q(\psi)\theta + \sum_{i=1}^{n} \log\left(p\left(y_i \mid \theta_i, \psi\right)\right)\right)
\end{aligned}$$

6.3.1 Approximate Bayesian inference with INLA

The objectives of Bayesian inference are the marginal posterior distributions for each element of the parameter vector

$$p\left(\theta_i \mid y\right)=\int p\left(\theta_i, \psi \mid y\right) \mathrm{d} \psi=\int p\left(\theta_i \mid \psi, y\right) p(\psi \mid y) \mathrm{d} \psi$$

and for each element of the hyperparameter vector

$$p\left(\psi_k \mid y\right)=\int p(\psi \mid y) \mathrm{d} \psi_{-k}$$

Thus, we need to perform the following tasks: (i) compute $p(\psi \mid y)$, from which also all the relevant marginals p(k|y) can be obtained; (ii) compute $p\left(\theta_i \mid \psi, y\right)$ which is needed to compute the parameter marginal posteriors $p\left(\theta_i \mid y\right)$

The INLA approach exploits the assumptions of the model to produce a numerical approximation to the posteriors of interest based on the Laplace approximation method introduced in Section 4.7 (Tierney and Kadane, 1986). The first task (i) consists of the computation of an approximation to the joint posterior of the hyperparameters as:

$$\begin{aligned}
p(\psi \mid y) &= \frac{p(\theta, \psi \mid y)}{p(\theta \mid \psi, y)} \\
&= \frac{p(y \mid \theta, \psi) p(\theta, \psi)}{p(y)} \frac{1}{p(\theta \mid \psi, y)} \\
&= \frac{p(y \mid \theta, \psi) p(\theta \mid \psi) p(\psi)}{p(y)} \frac{1}{p(\theta \mid \psi, y)} \\
&\propto \frac{p(y \mid \theta, \psi) p(\theta \mid \psi) p(\psi)}{p(\theta \mid \psi, y)} \\
&\approx \left.\frac{p(y \mid \theta, \psi) p(\theta \mid \psi) p(\psi)}{\tilde{p}(\theta \mid \psi, y)}\right|_{\theta=\theta^* \psi} =: \tilde{p}(\psi \mid y)
\end{aligned}$$

where $\tilde{p}(\theta \mid \psi, y)$ is the Gaussian approximation – given by the Laplace method – of $p(\theta \mid \psi, y)$ and $\theta^*(\psi)$ is the mode for a given ψ the Gaussian approximation turns out to be accurate since $p(\theta \mid \psi, y)$appears to be almost Gaussian as it is a priori dis- tributed like a GMRF, y is generally not informative and the observation distribution is usually well-behaved. The second task (ii) is slightly more complex, because in general there will be more elements in than in , and thus this computation is more expensive. A first easy possibility is to approximate the posterior conditional distributions $p(\theta \mid \psi, y)$ directly as the marginals from $\tilde{p}(\theta \mid \psi, y)$ i.e. using a Normal distribution, where the Cholesky decomposition is used for the precision matrix (Rue and Martino, 2007). While this is very fast, the approximation is generally not very good. The second possibility is to rewrite the vector of parameters as $\theta = (\theta_i, \theta_{-i})$ and use again Laplace approximation to obtain

$$\begin{aligned} p\left(\theta_i \mid \psi, y\right) &= \frac{p\left(\left(\theta_i, \theta_{-i}\right) \mid \psi, y\right)}{p\left(\theta_{-i} \mid \theta_i, \psi, y\right)} \\ &= \frac{p(\theta, \psi \mid y)}{p(\psi \mid y)} \frac{1}{p\left(\theta_{-i} \mid \theta_i, \psi, y\right)} \\ &\propto \frac{p(\theta, \psi \mid y)}{p\left(\theta_{-i} \mid \theta_i, \psi, y\right)} \\ &\approx \left.\frac{p(\theta, \psi \mid y)}{\tilde{p}\left(\theta_{-i} \mid \theta_i, \psi, y\right)}\right|_{\theta_{-i}=\theta_{-i}^*\left(\theta_i, \psi\right)} =: \tilde{p}\left(\theta_i \mid \psi, y\right) \end{aligned}$$

where $\tilde{p}\left(\theta_{-i} \mid \theta_i, \psi, y\right)$ is the Laplace Gaussian approximation to $p\left(\theta_{-i} \mid \theta_i, \psi, y\right)$ and $\theta_{-i}^*\left(\theta_i, \psi\right)$ is its mode. Because the random variables $\theta_{-i} \mid \theta_i, \psi, y$ re in general reasonably Normal, the approximation provided by (4.20) typically works very well. This strategy, however, can be very expensive in computational terms as $\tilde{p}\left(\theta_{-i} \mid \theta_i, \psi, y\right)$ must be recomputed for each value of and (some modifications to the Laplace approximation in order to reduce the computational costs are described in Rue et al., 2009).

Operationally, INLA proceeds as follows: (i) first it explores the hyperparameter joint posterior distribution $\tilde{p}(\psi \mid y)$ of Eq. (4.18) in a nonparametric way, in order to detect good points $\left\{\psi^{(j)}\right\}$ for the numerical integration required in Eq. (4.22). Rue et al. (2009) propose two different exploration schemes, both requiring a reparameterization of the ψ-space – in order to deal with more regular densities – through the following steps:

a) Locate the mode ψ^* of $\tilde{p}(\psi \mid y)$ by optimizing log $\tilde{p}(\psi \mid y)$ with respect to (e.g., through the Newton–Raphson method).

b) Compute the negative Hessian H at the modal configuration.

c) Compute the eigen-decomposition $= V\Lambda^{1/2}V'$, with $\Sigma = H^{-1}$.

d) Define the new variable z, with standardized and mutually orthogonal components, such that:

$$\psi(z) = \psi^* + V\Lambda^{1/2}z$$

The first exploration scheme (named grid strategy) builds, using the z-parameterization, a grid of points associated with the bulk of the mass of $\tilde{p}(\psi \mid y)$. This approach has a computational cost which grows exponentially with the number of hyperparameters; therefore the advice is to adopt it when K, the dimension of ψ, is lower than 4. Otherwise, the second explo- ration scheme, named central composite design (CCD) strategy, should be used as it reduces the computational costs. With the CCD approach, the integration problem is seen as a design problem; using the mode ψ^* and the Hessian H, some relevant points in the -space are selected for performing a second-order approximation to a response variable (see Section 6.5 of Rue et al., 2009 for details). In general, the CCD strategy uses much less points, but still is able to capture the variability of the hyperparameter distribution. For this reason it is the default option in R-INLA.

Chapter 7

Applications

Some *significant* applications are demonstrated in this chapter.

7.1 Example one

7.2 Example two

Chapter 8

Final Words

We have finished a nice book.

Bibliography

(2020). 7.2. advantages of using docker red hat enterprise linux 7.

(2020). An api generator for r.

(2020). Get docker.

(2020). Presentazione dei file robots.txt - guida di search console.

(2020). User agent: Learn your web browsers user agent now.

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., and Iannone, R. (2020). *rmarkdown: Dynamic Documents for R.* R package version 2.3.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2020). *shiny: Web Application Framework for R.* R package version 1.4.0.2.

Densmore, J. (2019). Ethics in web scraping.

Inc, F., Weststrate, M., Russell, K., and Dipert, A. (2020). *reactR: React Helpers.* R package version 0.4.3.

Lovelace, R., Nowosad, J., and Muenchow, J. (2019). *Geocomputation with R.* CRC Press.

Meissner, P. and Ren, K. (2020). *robotstxt: A 'robots.txt' Parser and 'Webbot'/'Spider'/'Crawler' Permissions Checker.* R package version 0.7.7.

Microsoft and Weston, S. (2020). *foreach: Provides Foreach Looping Construct.* R package version 1.5.0.

Perepolkin, D. (2019). *polite: Be Nice on the Web*. R package version 0.1.1.

Trestle Technology, LLC (2018). *plumber: An API Generator for R*. R package version 0.4.6.

Vaughan, D. and Dancho, M. (2018). *furrr: Apply Mapping Functions in Parallel using Futures*. R package version 0.1.0.

was previously an Economist at the Indeed Hiring Lab, A. F. F. (2020). Indeed tech skills explorer: Today's top tech skills.

Wikipedia (2020). Web scraping — wikipedia, l'enciclopedia libera. [Online; in data 15-luglio-2020].

Wikipedia contributors (2020). Cron — Wikipedia, the free encyclopedia. [Online; accessed 15-September-2020].

Wikiversità (2020). Scheduling — wikiversità,. [Online; accesso il 15-settembre-2020].

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1138700109.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.20.