Integrazione di una applicazione web con strumenti per la statistica

Franco Masotti

Università degli studi di Ferrara
Dipartimento di Matematica e Informatica

Relatore
Prof. **Fabrizio Riguzzi**

28 Settembre 2018

Università degli Studi di Ferrara

Obbiettivi

- L'obbiettivo della tesi è stato quello di permettere ad un ambiente web di programmazione logica di utilizzare un ambiente per la statistica.
- Vista la complessità di gestione dei software ho cercato di migliorare le loro modalità di installazione ed utilizzo.

Il linguaggio R

- R è un ambiente software orientato alla statistica e alla visualizzazione grafica. Comprende un'interpete dei comandi e un linguaggio di programmazione.
- Come tutti i linguaggi di programmazione, R mette a disposizione operatori, funzioni e strutture dati. I più importanti di questi sono:
  - L'operatore di assegnazione `<-`
  - La funzione di concatenazione `c`
  - Le liste
  - I data frame, un tipo particolare di lista
- Tutti questi elementi sono stati usati ampiamente.

Prolog, SWISH e Cplint on SWISH

- *Prolog* è un linguaggio di programmazione usato per esprimere fatti e regole attraverso relazioni logiche. *SWI Prolog* è un'ambiente completo e libero per la programmazione in Prolog.
- *SWISH* è un'applicazione web basata su SWI Prolog e viene usata per condividere codice. *Cplint on SWISH* è una particolare versione di *SWISH* che comprende strumenti basati sull'intelligenza artificiale.
- Alcuni programmi di Cplint on SWISH necessitano di grafici per la visualizzazione dei risultati. Grazie ad *R* è stato possibile disegnare i grafici.
- L'utilizzo di R è un'alternativa al sistema *C3.js* che è basato su Javascript ma genera grafici di qualità inferiore.

Scambiare i dati fra R e Prolog

- *rserve_client* è una libreria che permette l'accesso ad R da Prolog.
- *Rserve sandbox* è un server che lavora all'interno di un ambiente isolato chiamato *Docker*.
- Tutta la comunicazione fra client e server avviene attraverso un *socket di tipo UNIX*. Questo significa che il server è accessibile solo dalla macchina in cui gira così è garantita una migliore sicurezza.

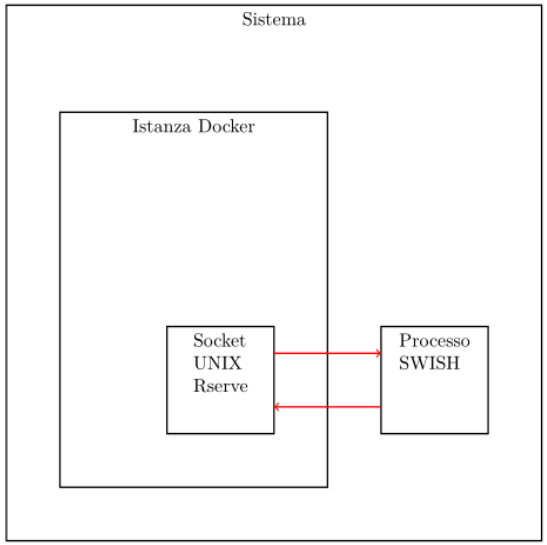

Figura: Schema funzionamento

QVM

- Vista la complessità e il numero degli strumenti utilizzati ho deciso di provare tutto all'interno di *macchine virtuali*, cioè software che permettono ad un sistema operativo di girare all'interno di un sistema ospite.
- Recuperando backup creati appositamente è infatti possibile partire da un ambiente sempre pulito senza perdere molto tempo.
- A questo proposito ho creato uno script shell, chiamato *QVM*, che permette la gestione di macchine virtuali attraverso il programma *QEMU*.

Pacchetti software

- Ho scritto alcuni *pacchetti software* che sono una serie di istruzioni che permettono l'installazione, la gestione e la rimozione dei programmi in modo semplice.
- Ho scritto i pacchetti per SWISH, Cplint on SWISH e per l'ambiente Rserve per le distribuzioni basate su *Arch Linux*, perché conoscevo già questo particolare sistema di pacchettizzazione.
- Per fare in modo che i pacchetti funzionino ho dovuto modificare alcune parti dei programmi originali.

ggplot2

- ggplot2 è un software R che permette di disegnare grafici usando la cosìddetta *grammatica dei grafici*.
- Anche se R mette già a disposizione librerie grafiche, ggplot2 è preferibile perchè la grammatica dei grafici permette di dividere il grafico per contesti diversi.
- Questo semplifica il lavoro dello sviluppatore e permette anche di trasformare i dati secondo scale e livelli diversi.

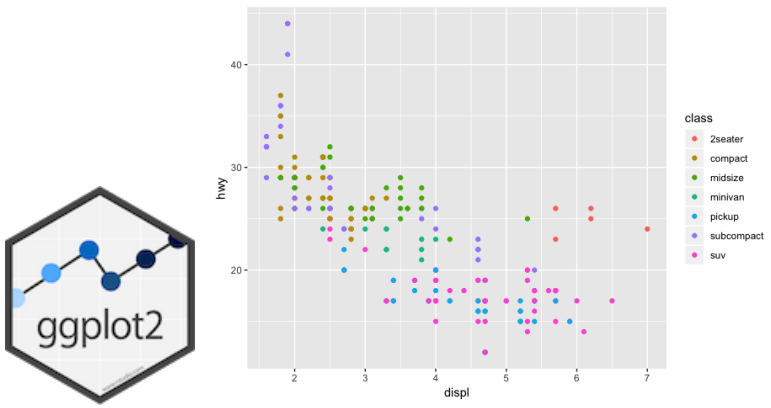

Figura: Logo ed esempio. Licenza GPLv2

Università degli Studi di Ferrara

cplint_r

- cplint_r è una liberia scritta in SWI Prolog da me che ha l'obbiettivo di graficare i risultati ottenuti da cplint usando la libreria ggplot2.
- La creazione di questa libreria è stata dettata dalla necessità di avere un'interfaccia uniforme e di evitare ripetizioni di codice.

```
/* Scale between 0 and 1 with 10 ticks (0.1,0.2,...,1)
 * This represents a probability between 0 and 1.
 */
geom_prob_bar(PTrue,PFalse) :-
    X=['T','F'],
    Y=[PTrue,PFalse],
    build_xy_list(X,Y,L),
    get_set_from_xy_list(L,R),
    r_data_frame_from_rows(df1, R),
    colnames(df1) <- c("names", "prob"),
    df <- data.frame(
        ids=as.character(df1$names),
        probabilities=c(df1$prob)
    ),
    <- ggplot(
        data=df,
        aes(
            x=ids,
            y=probabilities,
            fill=ids
        )
    ) + geom_bar(
        stat="identity",
```

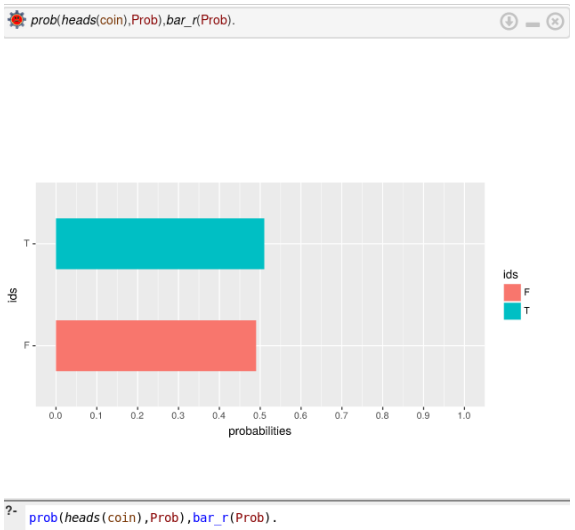

Figura: Estratto del codice di cplint_r e un esempio

Università degli Studi di Ferrara

Esempi

- In alcuni casi particolari non è stato possibile usare cplint_r perchè si trattava di grafici troppo specifici.
- Ad esempio ho modificato il programma kalman_filter in modo che nella versione R i due tipi di dati presenti siano separati in due sottografici. La versione C3.js infatti li raggruppa nello stesso grafico.
- Per il programma `gpr_R.pl` ho seguito il *working paper Gaussian Processes: A Quick Introduction*. Rispetto alla versione C3.js ho aggiunto e reso possibile la visualizzazione ed il calcolo della varianza e delle error bar.
- Si veda http://cplint.eu per una dimostrazione degli esempi.

Conclusioni

- L'obbiettivo di far comunicare R e Cplint on SWISH è stato raggiunto con cplint_r.
- L'obbiettivo di semplificare la gestione dei componenti software è stato raggiunto con i pacchetti.
- Esistono margini di miglioramento: l'aggiornamento *upstream* dei sorgenti sotware comporta il dover tener traccia di ogni loro cambiamento. Per questo potrebbe essere necessario modificare radicalmente i pacchetti software.