LAB : R Stampede

David Walling

Texas Advanced Computing Center
The University of Texas at Austin

July 12, 2013

Introduction

What you will learn

- How to perform basic data manipulation
- How to utilize 3rd party packages:
  - tm, caret, klaR, data.table, ggplot2
- How to profile code with Rprof
- How to parallelize code with mclapply
- How to use plots to analyze results
- General performance tips

What you will do

- Follow/modify example code READ the CODE COMMENTS
- Use R interactively with RStudio via VNC in 'vis' queue
- Submit R scripts as batch jobs to Stampede

Setup Lab Environment

Accessing Lab Files

- Log on to Stampede using your train## account.
- Untar the file lab_R.tar file (in ~train##).
- The new directory (lab_R) contains the scripts for the lab.
- cd into the appropriate subdirectory for an exercise.

You will be assigned this number.

```
$> ssh train##@stampede.tacc.utexas.edu
$> tar zxvf /work/00157/walling/lab_R.tar.gz .
$> cd lab_R
```

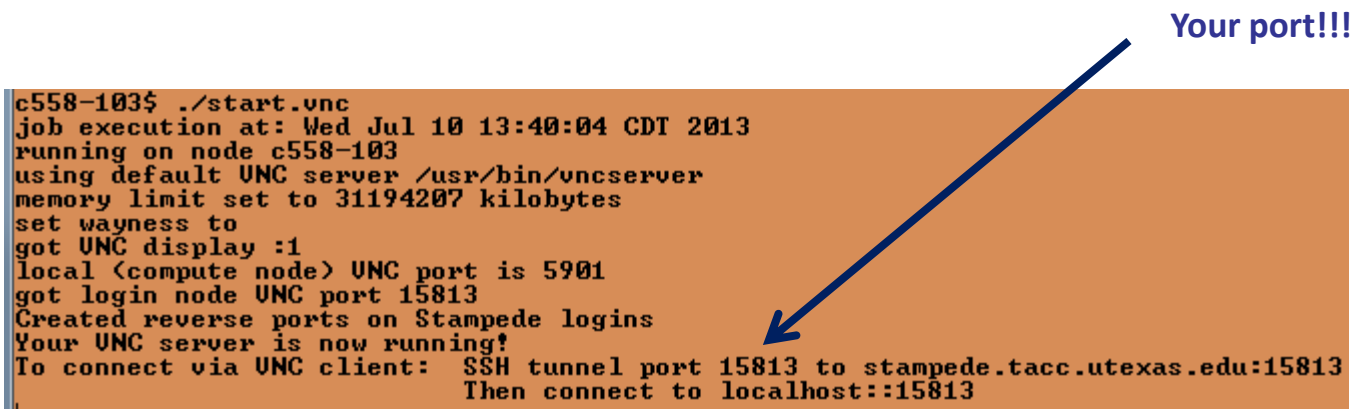
Install R Packages

- R packages will be installed in each users home directory.
- Must install from login node. Compute nodes cannot access CRAN.
- From login node:

```
$> cd lab_R
$> source R_sourceme.R
$> R
R> source("InstallPackages.R")
Select mirror 86
R> q()
No save
```

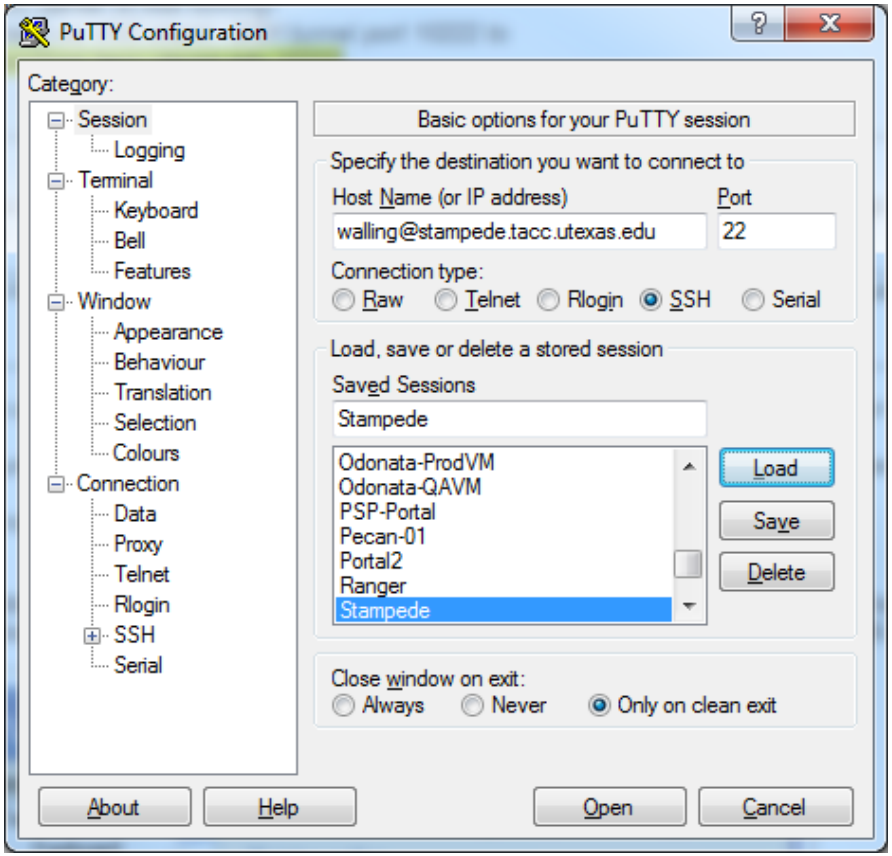
VNC Session

- Start VNC server
  - `$> cd lab_R`
  - `$> vim job.vnc`
  - `$> vncpasswd (REMEMBER!!!)`
  - `$> sbatch job.vnc`
  - `$> cat vncserver.out`

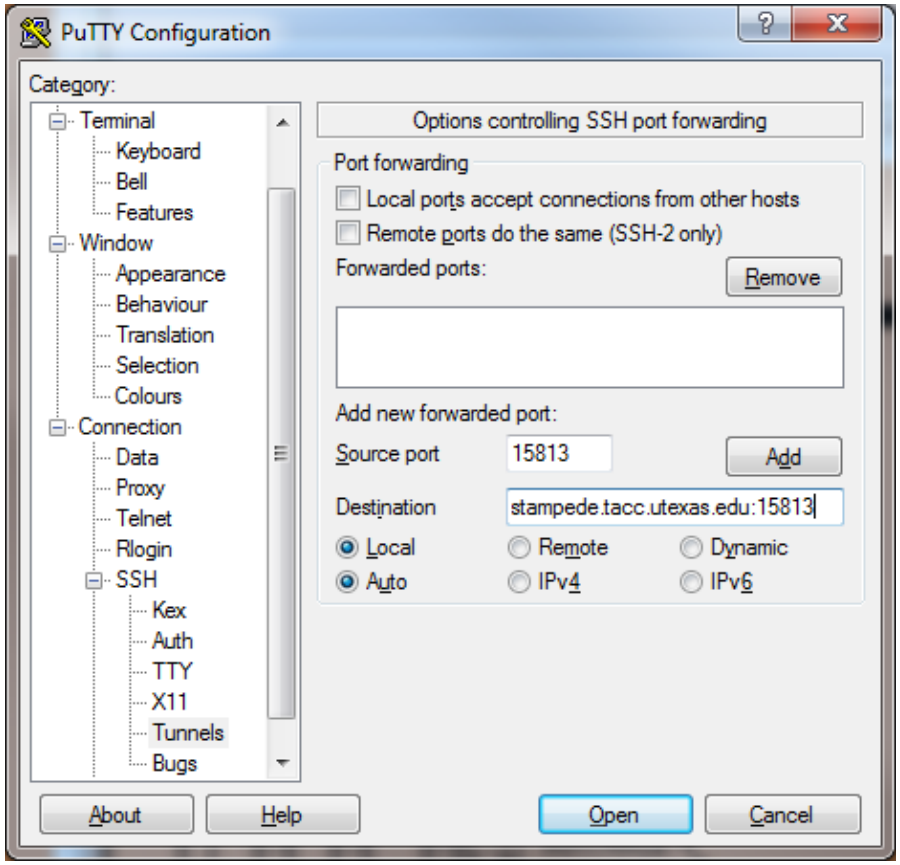


Windows SSH Tunnel (Putty)

Load Session



Create Tunnel



- Enter Source/Destination
- Click Add
- Click Open

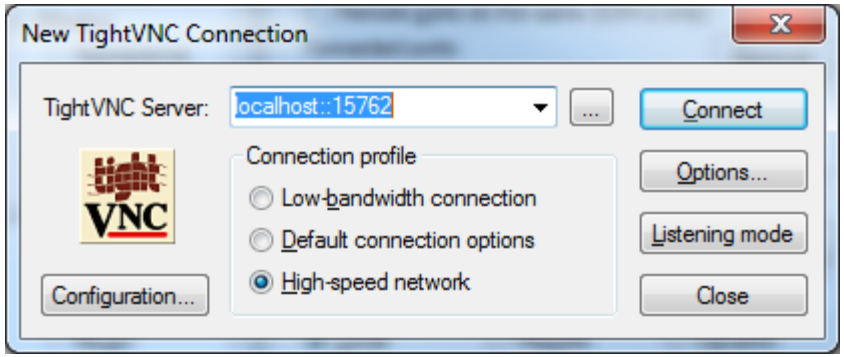
Linux/Mac SSH Tunnel

Create tunnel:

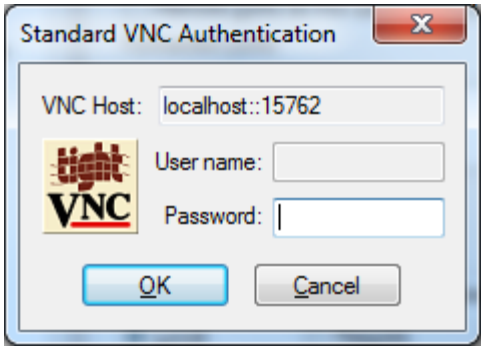
```
$> ssh -f -N -L xxxx:stampede.tacc.utexas.edu:yyyy
username@stampede.tacc.utexas.edu
```

VNC Client

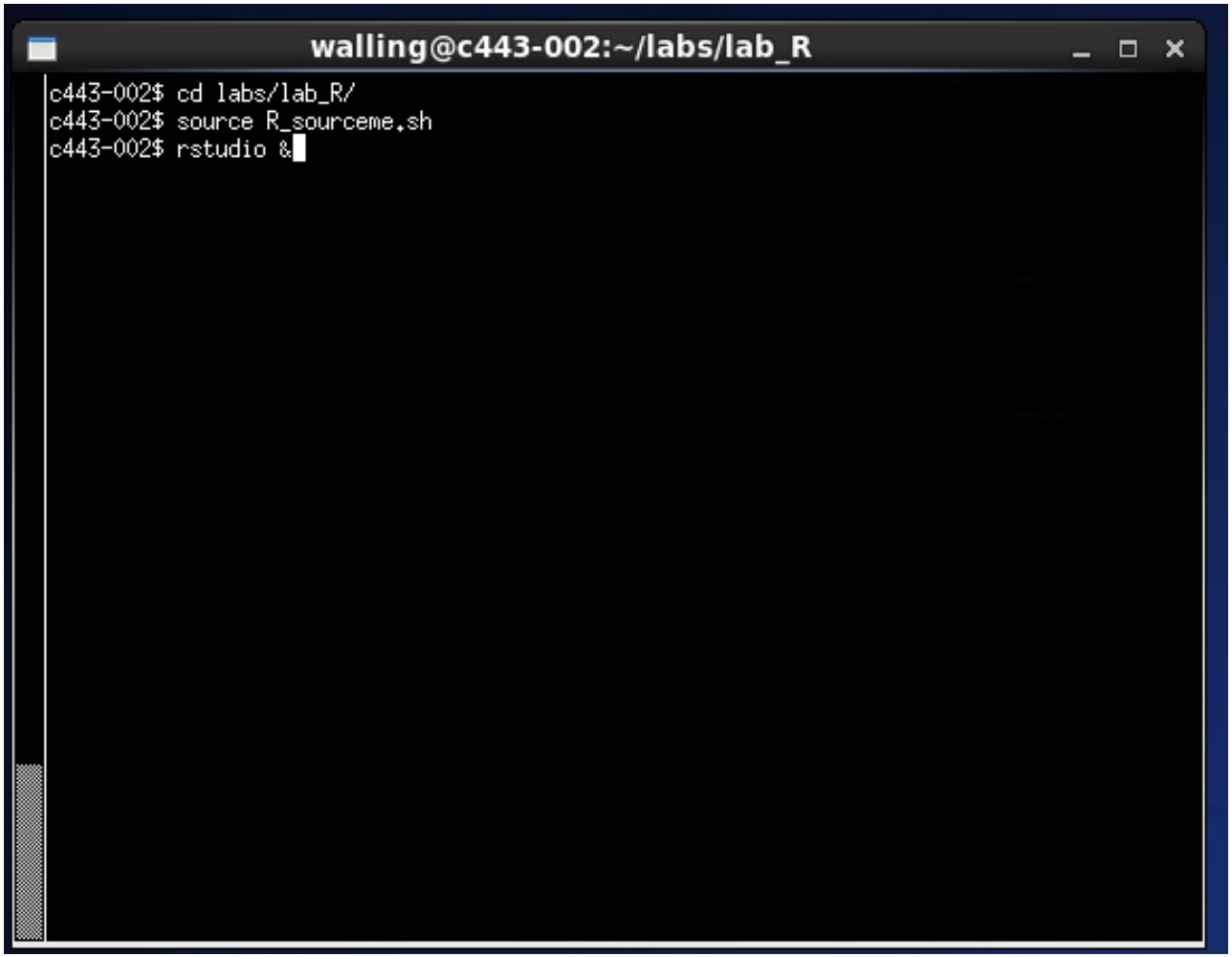
TightVNC



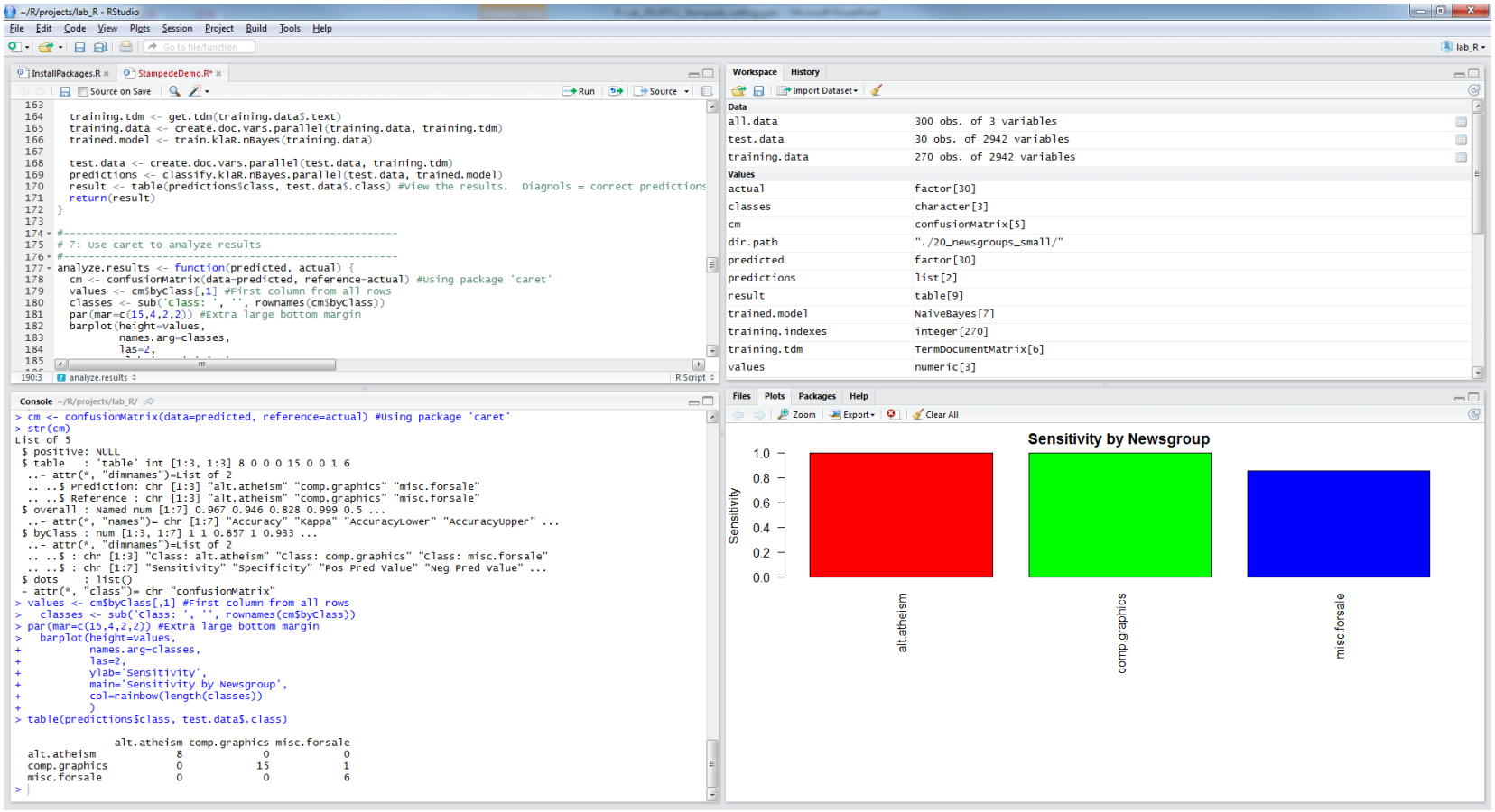
Enter vncpasswd



Open Project



RStudio



Note: currently cannot set –max-ppsize when using RStudio!!!

Building a Naïve Bayes Classifier

Raw Data

```
Path: cantaloupe.srv.cs.cmu.edu!crabapple.srv.cs.cmu.edu!fs7.ece.cmu.edu!europa.eng.gtefsd.com!gatech!asuvax!cs.utexas
From: joth@ersys.edmonton.ab.ca (Joe Tham)
Newsgroups: comp.graphics
Subject: Where can I find SIPP?
Message-ID: <yFXJ2B2w165w@ersys.edmonton.ab.ca>
Date: Mon, 05 Apr 93 14:58:21 MDT
Organization: Edmonton Remote Systems #2, Edmonton, AB, Canada
Lines: 11

        I recently got a file describing a library of rendering routines
called SIPP (SImple Polygon Processor).  Could anyone tell me where I can
FTP the source code and which is the newest version around?
        Also, I've never used Renderman so I was wondering if Renderman
is like SIPP?  ie. a library of rendering routines which one uses to make
a program that creates the image...

                                        Thanks,  Joe Tham

--
Joe Tham              joth@ersys.edmonton.ab.ca
```

Naïve Bayes

```
>   sample.data
     .file .class .text                boy   girl  fast  slow  ran   dog   want  love
[1,] 1     "Run"  "The boy ran fast."  TRUE  FALSE TRUE  FALSE TRUE  FALSE FALSE FALSE
[2,] 2     "Run"  "The girl ran slow." FALSE TRUE  FALSE TRUE  TRUE  FALSE FALSE FALSE
[3,] 3     "Run"  "He is fast."        FALSE FALSE TRUE  FALSE FALSE FALSE FALSE FALSE
[4,] 4     "Pet"  "I love my dog."     FALSE FALSE TRUE  FALSE FALSE TRUE  FALSE TRUE
[5,] 5     "Pet"  "I want a dog."      FALSE FALSE TRUE  FALSE FALSE TRUE  TRUE  FALSE
[6,] 6     "Pet"  "My dog is slow."    FALSE FALSE FALSE TRUE  FALSE TRUE  FALSE FALSE
```

Compute probabilities for each class:

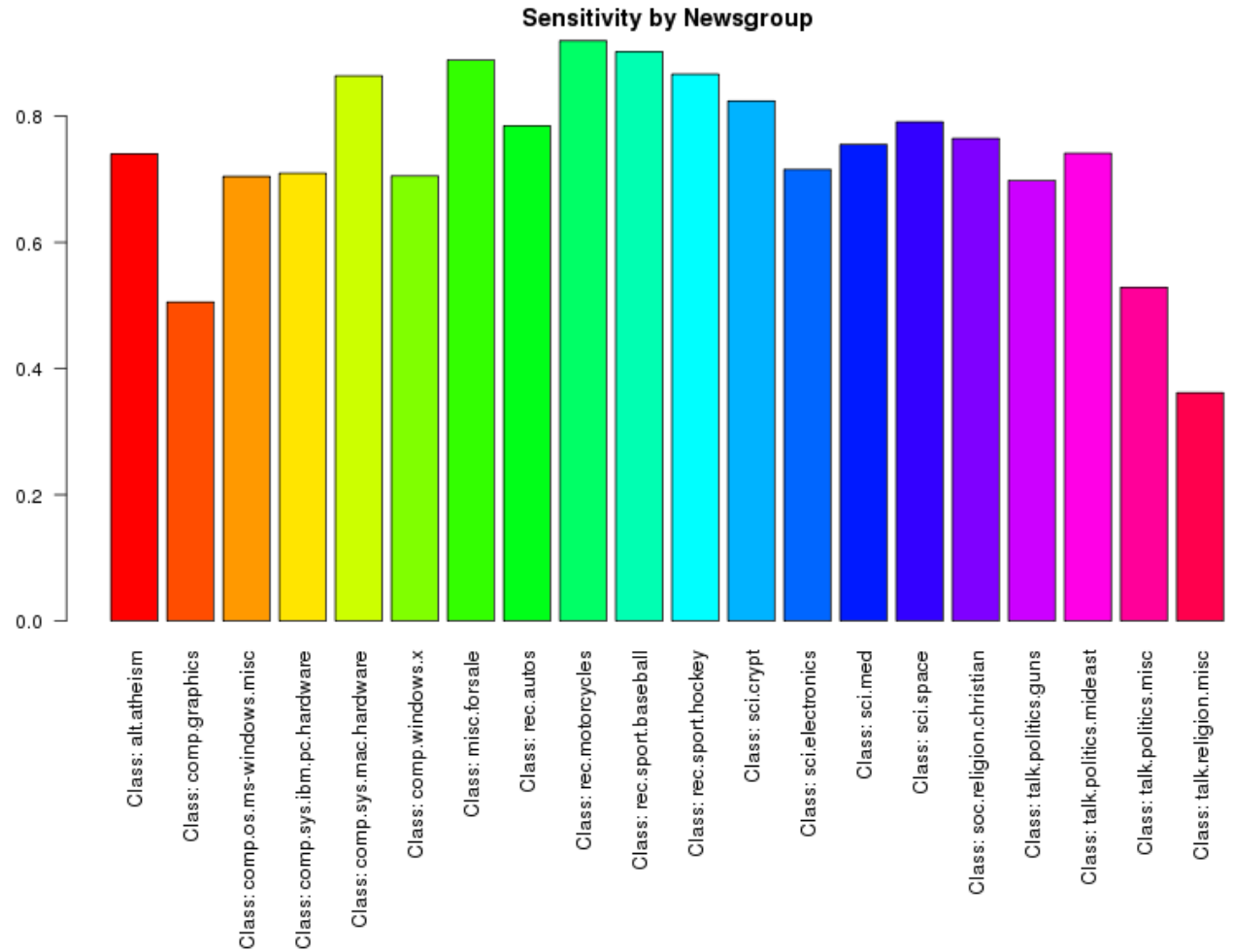
P(Pet | dog, slow) = P(dog, slow | Pet) * P(Pet) / P(dog, slow)
P(dog, slow | Pet) = P(dog | Pet) * P(slow | Pet) ***Assumes Independence!!!***
P(dog, slow) = P(dog) * P(slow)

P(Run | dog, slow) = P(dog, slow | Run) * P(Run) / P(dog, slow)
P(dog, slow | Run) = P(dog | Run) * P (slow | Run)
P(dog, slow) = P(dog) * P(slow) ***Same as above, we can ignore!!!***

Assign class with highest probability:

P(Pet | dog, slow) > P(Run | dog, slow)

Results


Sensitivity by Newsgroup
0.0
0.2
0.4
0.6
0.8
Class: alt.atheism
Class: comp.graphics
Class: comp.os.ms-windows.misc
Class: comp.sys.ibm.pc.hardware
Class: comp.sys.mac.hardware
Class: comp.windows.x
Class: misc.forsale
Class: rec.autos
Class: rec.motorcycles
Class: rec.sport.baseball
Class: rec.sport.hockey
Class: sci.crypt
Class: sci.electronics
Class: sci.med
Class: sci.space
Class: soc.religion.christian
Class: talk.politics.guns
Class: talk.politics.mideast
Class: talk.politics.misc
Class: talk.religion.misc

Code Walkthrough

- Create dataset from raw data files using basic R tools + *data.table/rbindlist* for speed
- Split dataset into train (90%) and test (10%) sets
- Add ‘term’ variables using *tm*
- Train our classifier using *klaR*
- Test our classifier and analyze results using *caret* and *ggplot2*
- Profile our code using *Rprof*
- Enhance using *parallel (mclapply)*

Performance Tips

- Avoid for loops! Use apply functions instead
- Only use data.frame when needed, i.e. have diff data types
- Use mclapply when inner code takes > 1min.
  - **Be careful of memory allocation!!!**
- Use optimized packages: data.table/fread/rbindlist, C implemented code, etc…
- Be aware of subtle diffs in similar functions: data.matrix vs as.matrix
- Search for existing code before writing your own
- Develop against subsets of data to get everything working
- Profile code at small/medium scale to quickly identify bottlenecks

Batch Script

rJob.slurm

```
#!/bin/bash
#
#----------------------------------------------------------------------------
# An example script for launching an R job in batch mode
#----------------------------------------------------------------------------
#
#SBATCH -J myRJob                   # Job name
#SBATCH -o myRJob.out               # Name of stdout output file (%j expands to jobId)
#SBATCH -p normal                         # Queue name
#SBATCH -N 1                         # Total number of nodes requested (16 cores/node)
#SBATCH -n 1                        # Total number of mpi tasks requested
#SBATCH -t 04:00:00                  # Run time (hh:mm:ss) - 4 hours
#SBATCH -A TACC-DIC
#SBATCH --mail-user=walling@tacc.utexas.edu
#SBATCH --mail-type=begin  # email me when the job starts
#SBATCH --mail-type=end    # email me when the job finishes

module load R
Rscript myScript.R
```