MLA class 1

Mariano Dominguez

June 29, 2015

Some news about Astrostatistics:

Name of the new Commission: Astroinformatics and Astrostatistics

Rationale The Rise of Statistics and Informatics in Astronomy

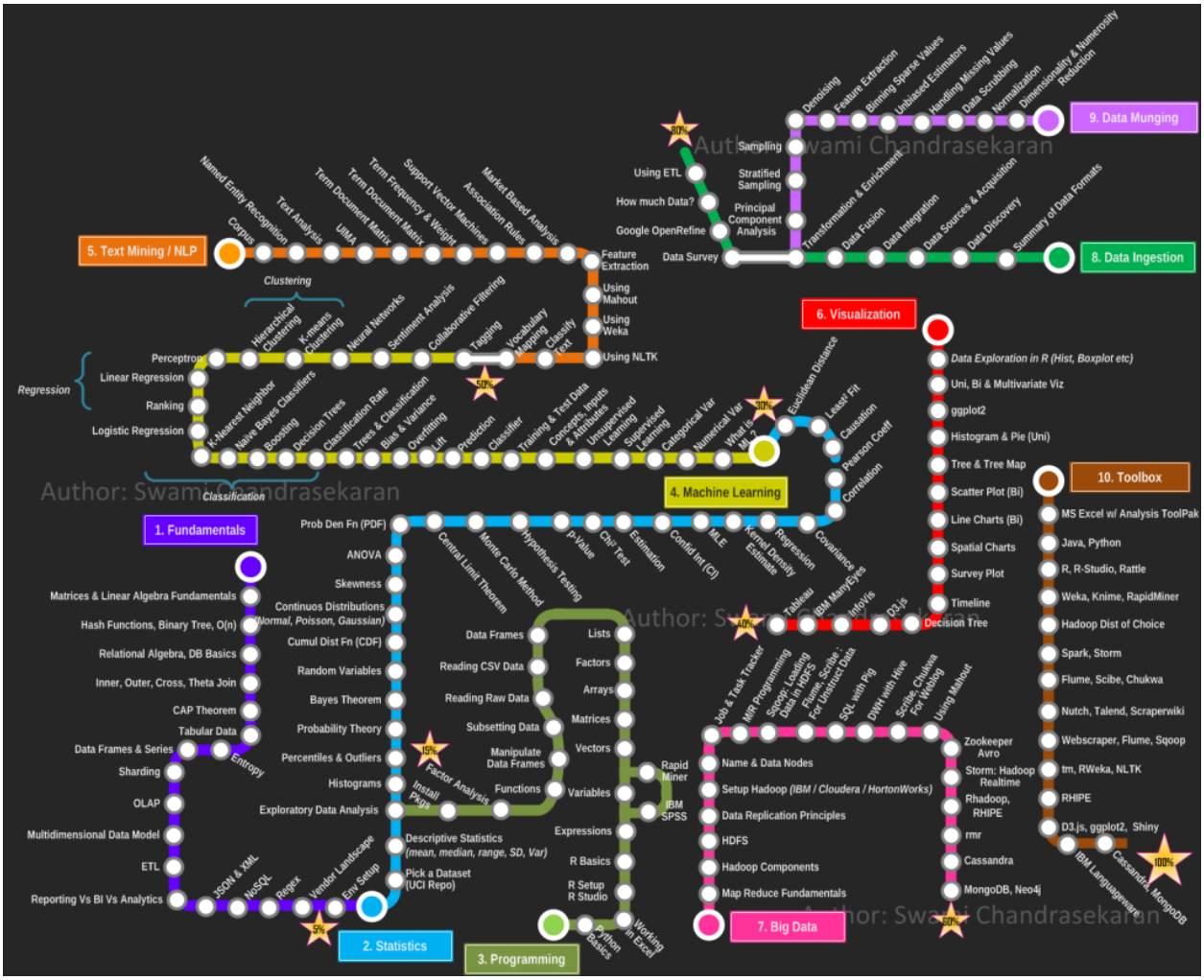
Astronomy is entering the cyber age. Astrostatistics and astroinformatics are burgeoning enterprises with rapid growth in the astronomical research literature, cross-disciplinary meetings, new textbooks, and attention by scholarly societies. These trends are responding to the proliferation of large-scale imaging, spectral and time domain data. Advanced computational and statistical analysis techniques are essential to analyzing our increasing data flow and realizing our endeavor to characterize and understand cosmic phenomena. We face growing challenges in interpretation of large datasets, such as mapping Dark Matter from subtle weak lensing signals and finding Type Ia supernovae to measure Dark Energy. But we also tackle intricate statistical inferential problems with small datasets, such as Bayesian model selection in multi-planet systems and analysis of irregularly sampled time series. These problems arise in all fields of astronomy – solar, planetary, stellar, Galactic, extragalactic and cosmological studies.

Astronomers are making rapid progress, but the broader fields of statistical inference, data mining, and high performance computing are constantly changing. An IAU Commission on Cyber-Astronomy (CCA) devoted to these issues is needed to improve our methodology, training, and interactions within and outside our discipline. While the urgency of these issues is increasing, few astronomers have professional-level training in the foundations and recent advances in statistics, computer science, and information technology. Cross-disciplinary interaction with experts in allied fields is still nascent but has great promise.

The Astrostatistics and Astroinformatics Portal (http://asaip.psu.edu)

is a new Web site serving the cross-disciplinary communities of astronomers, statisticians and computer scientists. It is intended to foster research into advanced methodologies for astronomical research, and to promulgate such methods into the broader astronomy community.

- The comunities involved ranged so far
- Machine Learning, Statistical Learning, Artificial Intelligence, Multivariate Statistics.
- Data Mining, Database, Big Data, Computer Vision, Vizualization, Information Theory.



Check the phenomena by yourself:

http://www.google.com/trends/explore#q=%2Fm%2F0c_xl%2C%20data%20scientist%2C%20computer%20scientist%2C%20data%20analyst&cmpt=q&tz=Etc%2FGMT%2B3

the rapid development of these fields over the last few decades was led by by advance in scientific computing:

- improved hardware technology (supercomputing, HPC)
- open source movement (Apache Spark, postgresSQL, see http://opensource.org/licenses/alphabetical)
- non-sql and new-sql databases (Hadoop, SciDB)
- data analysis lenguage R (Python, Julia)
- parallelism (mpi, open-mp, cuda, openCL)
- the cloud (github, overleaf, shinny, plotty, VOs)
- sources of big data (astronomy, genomics, social networks, internet of things)
- new (old) machine learning algorithms (deep convolutional neural networks, adapboost, WEKA)
- a growing number of Data Science courses, master, doctorates. (http://datascience.nyu.edu/, http://cd3.caltech.edu/, http://idies.jhu.edu/, etc)
- there is gold in the data! (https://www.kaggle.com/ ,data challenge etc)

We will consider on this course: techniques in all these areas wich are most often applied in the analysis of astronomical data.

Data Mining (knowledge discovery) is a set of techniques for analyzing and describing structured data, for example finding patterns in a large dataset. Common methods include:

- density estimation,
- unsupervised classification
- clustering
- principal component analysis

From this point of view is not important contrast these data trends with a model. In short data mining is about what the data themselves are telling us. This is what statisticians call *exploratory data analysis*.

Machine Learning (Statistical Learning) is an umbrella term for a set of techniques for interpreting the data by comparing them to models. Common methods include:

- regression methods
- maximum likehood estimators
- Bayesian methods They are often called Inference techniques, data based statistical inferences or just plain old *fitting*.

You should select a problem(s) in astronomy that could be solved using these techniques in order to apply them in the practical part of this course.

Machine learning in ussually divided into two main types. In the **predictive** or **supervised learning** approach, the goal is to learn a mapping from inputs

$$\vec{x}$$

to outputs y, given a set of labeled set of N input-output pairs

$$D = (\vec{x_i}, y_i)$$

. Here D is called the **training set**, and N is the number of training examples.

In the simplest setting, each training input x_i is a D-dimensional vector of numbers. These are called **features, attributes** or **covariates**. In general, however x_i could be a complex structure object, such as an image, a sentence, an e-mail message, a time series, a molecular shape, a graph etc.

Similarly the form of the output or **response variable** can be in principle anything, by most methods assume that y_i is **categorical** or **nominal** variable from some finite set, or that is a **real-valued scalar**. In the former case, the problem is known as **classification** or **pattern recognition** and in the last is known as **regression**. Another variant, known as **ordinal regression**, occurs when label space has some natural ordering, such as grades A-E.

The second main type of machine learning is the **descriptive** or **unsupervised learning** approach. Here we hace only given inputs D and the goal is to find "interesting patterns" in the data. This is sometimes called **knownledge discovery**. This is a much less well-defined problem, since we are not told what kind of patterns to look for, and there si no obvius error metric to use (unlike supervised learning).

There is a third type of machine learning, known as **reinforced learning**, which si somewhat less commonly used. This is useful for learning how to act or behave when given occasional reward or punishment signals (for example, consider how a baby learns to walk, or see http://arxiv.org/abs/1312.5602).

The prototypical application could be seen here: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/ and http://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer

---

Introduction to the R Statistical Lenguage http://r-project.org and CRAN http://cran.r-project.org/ :

R consists of a collection of software for infrastructure analysis, and about 25 important packages providing a variety of important data analysis, applied mathematics, statistics, graphics and utilities packages. The CRAN add-on packages are mostly supplied by users, sometimes individual experts and sometimes significant user communities in biology, chem- istry, economics, geology and other fields.

There is a nice IDE: RStudio https://www.rstudio.com/ and graphics library ggplot2 (now ggviz), try it! Also exists several R graphs gallery on internet. Remember that recently was stablished the R consortium: https://www.r-consortium.org/ (Microsoft buys Revolution, also see h2o)

Considerable effort has been made to connect R to other programs, languages and analysis systems. External R scripts can be easily run in the console using the source function, but more effort is needed to run programs in other languages. As listed in Table B.4, R connects to BUGS (Bayesian inference Using Gibbs Sampling), C, C++, FORTRAN, Java, JavaScript, Matlab, Python, Perl, XLisp and Ruby. In some cases, the interface is bidirectional allowing R functions to be called from foreign programs and foreign programs to be called from R scripts.

Introduction to the Python Lenguage and the git code management tool:

Python is open source, object oriented lenguage with a well developed set of libraries , packages and tools for scientific computation.

The core packages for scientific computing are:

- Numpy
- SciPy
- Matplotlib
- astroML
- Scikit-learn
- PyMC
- Healpy
- Astropy
- Pandas

There are two main private vendors: Enthought (https://www.enthought.com/products/canopy/) and Continuum Analytics (https://store.continuum.io/cshop/anaconda/), and a good number of editors offer python support (Spyder)

One ring to rule them all:

Is strongly recomended to install the Ipython, R and Julia development tool http://jupyter.org/

and the git code management tool. There are other freely available similar tools like CVS, SVN, Mercurial etc with basic similar functionality. They support collaborative development of software and tracking of changes to software source code over time.

One of the most useful features of Git is the ability to set up a remote repository, so that code can be checked in and out from multiple computers. Even when a computer is not connected to a repository, the local copy can be still be modified and changes reported to the repository latter.

Read the document in https://guides.github.com/introduction/flow/index.html

Computational science must develop standards for reproducibility before it can be considered a third (fourth) branch of the scientific method i.e. Reproducible Research or Data and Code Sharing with publication.

- An incomplete survey of the relevant literature:
- Numerical Recipes 3 by Press, Teukolsky, Vetterling and Flannery.
- The Elements of Statistical Learning: Data Mining, Inference and Prediction by Hastie, Tibshirani and Friedman.
- Modern Statistical Methods for Astronomy with R Applications by Feigelson and Babul.
- Information Theory, Inference and Machine Learning Algorithms by Mac Kay.
- Pattern Recognition and Machine Learning by Bishop.
- Artificial Intelligence by Russell and Norvig.

You can embed an R code chunk like this:

```
library(C50)
library(printr)
```

This code takes a sample of 100 rows from the iris dataset:

```
train.indeces <- sample(1:nrow(iris), 100)
iris.train <- iris[train.indeces, ]
iris.test <- iris[-train.indeces, ]
```

This code trains a model based on the training data:

```
model <- C5.0(Species ~ ., data = iris.train)
```

This code tests the model using the test data:

```
results <- predict(object = model, newdata = iris.test, type = "class")
```

This code generates a confusion matrix for the results:

```
table(results, iris.test$Species)
```

results/	setosa	versicolor	virginica
setosa	16	0	0
versicolor	0	15	4
virginica	0	1	14