Statistical learning and distributed computing

Joannès Vermorel
February 10th, 2006

Center of Bioinformatics, École des Mines

Summary

- Overviews
  - Statistical learning
  - Distributed computing
  - Software engineering
- A big picture of computational biology
- A synthetic approach
- PhD works
  - Algorithmic toolbox for distributed computing
  - Programming framework for distributed computing
  - Distributing statistical learning algorithms
- Conclusions – Future works

Statistical learning overview

- Intuitive definition
  - Using a list of `(input, output)` items.
  - Build a function `input` → `output`.
- Ex: given 1,000 proteins whose toxicology is known, can we predict the toxicology for the 1,000,000 remaining proteins ?
- Statistical learning aims to provide knowledge/tools
  - as general as possible.
  - as accurate as possible.

Distributed computing overview

- Intuitive definition
  - with N computers available.
  - perform a computation N times faster.
- Ex: no single machine (1 processor, 1 disk) can crawl the whole web and provide an efficient search engine.
- Distributed computing aims to provide knowledge/tools
  - as general as possible.
  - as scalable as possible.
- *Perspective note: even cell-phones are likely to be distributed machines in 5 years.*

Software engineering overview

- Intuitive definition
  - all tools that increase developer productivity.
  - all methods that increase developer productivity.
- Ex: programming languages are the consequence of the need for productivity.
- Software engineering aims to provide knowledge/tools
  - as general as possible.
  - as productive as possible.
- *Perspective note: fuzzy topic, hard to quantify anything, but huge progresses in the last decade.*

Big picture of Computational Biology

- The amount of biological data vastly exceed what a human expert can handle.
  - Ensembl Trace database (DNA sequences) is 22TB large.
- All future developments in biology will heavily rely on computational tools.
- Statistical learning tackles many major issues for biology and the pharmaceutical industry.

A synthetic approach

- The amount of biological data increases at a tremendous rate.
- Ex: Ensembl Trace database has been doubling in size every 10 months.

- Consequence: data becomes too large to be processed on a single machine.
- Memory and CPU hardware units do not match the data growth.

→ Need for distributed computing.

A synthetic approach

- And software engineering?
  - “Computational biologists” (at least in the CB) spends more than half of their time designing software.
  - The “software” complexity of statistical learning methods increases quickly.
  - Efficiently (and reliably) distributing an algorithm is a challenging task.
- Software engineering is needed to “scale-up” computational biology research.

PhD research works

- Algorithmic toolbox for distributed computing.
  - Data streams
  - Near neighbor search
  - Multi-armed bandit
- NGrid, a programming framework for distributed computing
  - project overview
  - distributed garbage collector
- Distributed machine learning algorithms
  - clustering
  - support vector machines

Algorithmic toolbox

- Idea: gathering the basic pieces usually required to distribute machine learning algorithms.
- Goals
  - (Distributed) computing performance.
  - Developer productivity.
- Requirements: generality.

Algorithmic toolbox Data Streams

- Data stream algorithms
  - Huge list of items.
  - Read one-item-at-a-time model.
  - Get a statistical measure over the whole list.
- Ex: AT&T, get the median TCP/IP packet size on a router carrying 100G packets a day.
- Classical data stream algorithms
  - Median / Quantile computations.
  - Many “online” versions of “offline” algorithms.
- My contribution
  - Online histograms (approximation of an online distribution) with Hervé Bronnimann (Polytechnic University).

Algorithmic toolbox
Near Neighbor Search

- Near neighbor search algorithms
  - A set of data points.
  - A measure of similarity between the points.
  - How to get the neighbors of a query points?
- Ex: AT&T, near neighbors where used to detect fraud (using similarity between customer profiles).
- Classical solutions rely on the *triangular inequality* assumption.
- My contribution
  - Research report “Near neighbor search in non-metric space”.
  - More precise performance measurements (rank based criterion).
  - The triangular inequality has virtually no impact on performance.
- Future works: improve performance based on those insights.

Algorithmic toolbox Multi-Armed Bandit

- Multi-Armed bandit
  - A set of levers.
  - Unknown rewards associated to each lever.
  - Maximize the sum of rewards in an iterative play.
- Ex: AT&T, automated customer care, dialog system was taking initiatives, the rewards were associated with customer feedback.
- Classical solutions were purely theoretical (no empirical evaluation).
- My contribution
  - Paper “Multi-Armed Bandit Algorithms and Empirical Evaluation”, ECML’05, with Mehryar Mohry (New York University).
  - Large benchmark of the literature.
  - New algorithm POKER, perform better than known solutions.

NGrid
Dist. computing framework

- A distributed computing framework is essentially a tradeoff between
  - Framework performance overhead.
  - Framework expressivity.
  - Development productivity.
- NGrid targets machine learning algorithms.
  - project started August 2005.
- Most similar projects
  - MapReduce (Google).
  - ProActive (Inria).

NGrid
Dist. computing framework

- NGrid is
  - open source, LGPL, `http://ngrid.sourceforge.net`
  - implemented in .Net / C#.
- NGrid key elements
  - just a “library”, not a new language.
  - distributed objects.
  - distributed threads.
  - distributed garbage collection.

NGrid
Dist. computing framework

- Distributed Garbage Collection (DGC)
  - Essential for developer productivity.
  - Critical for software reliability.
- Research paper (submitted)
  - “Sketch-based distributed garbage collection”
  - Insight: using data stream methods to reduce the bandwidth requirements for the DGC.
- Future directions: combining data streams and machine learning to perform smart load balancing with NGrid.
  - load balancing → generalized multi-armed bandit problems.

Distributed Statistical Learning

- Most recent part of my work
- Very preliminary results

Dist. stat. learning Clustering

- Clustering task (naïve yet most frequently used data mining operation)
  - Large dataset.
  - Measure of similarity between dataset items.
  - Find “clusters” (i.e. groups) of similar items within the dataset.
- Half a dozen of heuristics
  - based on simple machine learning.
  - improving the speed of the mono-threaded clustering algorithm.
- Distributed clustering
  - heavily rely on near neighbor search algorithms.
  - Distributed clustering algorithm fits perfectly the distributed object framework of NGrid.

Dist. Stat. Learning Support Vector Machines

- Support Vector Machines (SVM)
  - One of the most successful methods in machine learning.
  - Wide range of applications.
- SVM performance issues
  - memory requirements are supra-linear.
  - CPU requirements are supra-quadratic.
- Distributing SVM is a challenging task
  - Known SVM algorithms are deeply iterative.
- Future works: approximate the SVM algorithms
  - Enable distributed approach.
  - Highly probable: the solution will rely on clustering algorithm.

Conclusions
Future works

- Mastering several domains is tough
  - Yet the situation is mostly inescapable to perform large scale statistical learning.
- Strong interactions between those domains
  - Each domain benefits from the advances of the others.
- Future works
  - Bringing NGrid at an operational level.
  - Implementing machine learning algorithms on top of NGrid.