Speaking Maths

Marc Toussaint

October 14, 2015

Systems can be described in many ways. Biologists describe their systems often using text, and lots and lots of data. Architects describe buildings using drawings. Physisics describe nature using differential equations, or optimality principles, or differential geometry and group theory. The whole point of science is to find descriptions of systems—in the natural science descriptions that allow prediction, in the synthetic/engineering sciences descriptions that enable the design of good systems, problem-solving systems.

And how should we describe intelligent systems? Robots, perception systems, machine learning systems? I guess there are two main options: the imperative way in terms of literal algorithms (code), or the declarative way in terms of maths.

The point of this lecutre is to teach you to *speak maths*, to use maths to describe systems or problems. I feel that most maths courses rather teach to consume maths, or solve mathematical problems, or prove things. Clearly, this is also important. But for the purpose of intelligent systems research, it is essential to be skilled in expressing problems mathematically.

If you happen to attend a Machine Learning or Robotics course you'll see that *every* problem is addressed the same way: You have an "intuitively formulated" problem; the first step is to find a mathematical formulation; the second step to solve it. The second step is often technical. The first step is really the interesting and creative part. This is where you have to nail down the problem, i.e., nail down what it means to be intelligent/good/well-performing, and thereby describe "intelligence"—or at least a tiny aspect of it.

The "Maths for Intelligent Systems" course will recap essentials of linear algebra, optimization, probabilities, and statistics. These fields are essential to formulate problems in intelligent systems research and hopefully will equip you with the basics of speaking maths.

1 Seeming trivialities about notation

Equations and mathematical expressions have a syntax. This is hardly ever made explicit[1] and might seem trivial. But it is surprising how buggy mathematical statements can be in scientific papers (and exams). I don't want to write much text about this, just some bullet points:

- Always declare objects. Be aware of variable/index scoping. Type checking..

[1] Except perhaps by Gödel's incompleteness theorems and areas like automated theorem proving.

- $\arg\min$
- sets, tuples, $\{x_i\}_{i=1}^n$, $(x_i)_{i=1}^n$, $(x_1, .., x_n)$, $x_{1:n}$
- set notations $\{f(x) : x \in \mathbb{R}\}$, $\{n \in \mathbb{N} : \exists\ \{v_i\}_{i=1}^n$ linearly independent, $v_i \in V\}$
- $(a, b) \in A \times B$,
- outer product $x \otimes y$ (slightly different in standard vector; and tensor context)
- cross product
- direct sum $\oplus$
- Decorations are ok, but really not necessary. It is much more important to declare all things. E.g., there are all kinds of decorations used for vectors, $\boldsymbol{v}, \underline{v}, \overrightarrow{v}, |v\rangle$ and matrices. But these are not necessary. Properly declaring all symbols is much more important.
- Never use multiple letters for one thing. E.g. $length = 3$ means l times e times n times g times t times h equals 3.