Machine Learning in Secondary Education?

Ricard Gavaldà GAVALDA@LSI.UPC.EDU
Universitat Politècnica de Catalunya (UPC), Barcelona (Spain)

Abstract

Given the alarming drop in applications to Computer Science (CS) studies across Europe, any efforts at presenting CS as an interesting, challenging, and useful discipline are welcome and necessary. In this note I present my view that it is possible to introduce motivated secondary-school students to the basic goals, techniques, and applications of Machine Learning. Furthermore, I claim that Machine Learning is almost unique among the subfields of CS with this property, and therefore the Machine Learning community has a special duty to help attracting students to CS. I then present some very preliminary thoughts on a project aiming at developing and deploying appropriate teaching materials, and argue that the PASCAL2 NOE is the ideal agent to set up and coordinate such an initiative in Europe.

An even more preliminary version of this note was presented at the *2008 Teaching Machine Learning Workshop* in Saint-Étienne, France.

1. The Proposal

In this note, I discuss the possibility and convenience of creating teaching materials introducing the basics of Machine Learning to high-school students.[1] Obviously, I am not thinking of including these contents as mandatory or taking up a whole course. Rather, I have in mind short, optional modules that instructors could include and combine within larger courses (e.g., a few sessions or weeks within yearly mathematics, technology, or IT courses), probably only to a subset of interested students.

Concerning the possibility of doing so: Machine Learning is a quite unique subfield of Computer Science[2] having the three following features: 1) The basic techniques can be explained from the mathematics taught in high school (e.g. vectors and vector arithmetic, linear algebra, probability). 2) Hands-on experience is possible without knowledge of programming, with results that the students can interpret. And 3) Useful, realistic applications and research challenges can be described – appealing, if necessary, to the existence of more advanced techniques than those presented in the course. Also, technologies exist to deploy easily and cheaply a variety of suitable teaching resources in high schools, and even at home for interested students.

Concerning the interest of the proposal: As is well known, enrolment figures in Computer Science are decreasing to socially dangerous levels in most of Europe and North America. There is a variety of demographic, sociological, and academic reasons for this, but for sure part of the problem is the social image of the field. All kinds of efforts should be made to (honestly) present it to prospective students as intellectually challenging, professionally exciting, and socially useful, and again Machine Learning offers ample opportunities in this respect.

On the teacher's side, it is my general experience that high-school instructors spend a great deal of their time looking for ideas, resources, and activities for their teaching, and welcome materials that can be used without much effort.

To end up, I present some very preliminary thoughts on a possible "Machine Learning for Secondary Education" project, the main difficulties it could present, and why the PASCAL2 NOE could be the leader of such an initiative within Europe.

[1] I use indistinctly the terms secondary education and high-school education to denote the 2 to 4 years preceding university-level education. Of course, names, structure, and contents of this educational level vary across countries.

[2] I very lazily use the name "Computer Science" or CS to mean a large number of academic disciplines and curricula that are organized in very different ways across countries. "Informatics" is, in my opinion, a far better name, though not as widespread.

2. Context

2.1. Secondary Students and CS

The last 4-5 years have seen a steady decrease in the quantity of applications to university-level CS studies in the most of Europe and North America. Partly (but only partly) because of this, the average quality of applicants seems to be lowering too: good students that would have chosen CS a few years ago seem to be choosing other studies now. If this trend persists, it poses a very serious threat to the growth of IT industry in Europe and, therefore, a problem for society as a whole.

There are many factors explaining this drop, and a detailed analysis or diagnostic is out of the scope of this note (and of my ability). It is safe to say, however, that important factors are a series of damaging and common social (mis)conceptions about CS and CS-related professional careers. Some examples are: Computing professionals have less social "prestige" than other professionals (doctors, architects, other types of engineers, etc.); CS professionals work long hours in stressing but boring projects; after the dot-com bubble, the career market is not expanding anymore, and jobs are unstable and underpaid, etc.

Without discussing truth or falsity of these beliefs, I want to focus on one in particular which, in my opinion, is driving away many good students with a scientifically oriented mind: the conception that CS as a subject deals mostly with routine business applications, or, at best, with compulsory, hacker-type, programming. There is no real challenge remotely comparable to those found in mathematics, medicine, biology, or physics. Indeed, how can CS provide excitement like that of understanding how the cell works, creating the cure for cancer, or finding the traces of the Big Bang in deep space?

2.2. CS in High School

Nowadays, most primary and high-school students in the EU area do get (or should get in the next decade) the basic computing skills that all citizens will be pretty soon expected to have (writing documents, using spreadsheets or similar tools, organizing information and searching it in the web, etc). These skills are acquired either in specific IT courses or instrumentally within other courses.

Many schools offer interested students further optional modules introducing "real computing", namely… programming!! Students learn the syntax of Java or C and use them to write a few programs (even surprisingly complex ones); they learn how to download and install programs (if they have not learned this by themselves before); and they learn to create and query databases using SQL. If the school is large, and there are enough interested students, and a sufficiently passionate teacher is around, even a further module may be offered with advanced material: students get to write Java or C programs *with a GUI,* to install and manage *a whole Linux,* and to create databases *that can be queried via web using PHP.*

This effort, commendable as it is, transmits the student the idea that CS equals programming, and that the challenges in CS are writing larger and larger programs and managing bigger and bigger systems. This does not necessarily make the discipline attractive, especially if the potential applications of such larger programs and systems are not explained at the same time.

Also, in most cases the teachers for these courses have had no formal training on CS themselves. In this respect, initiatives that aim at familiarizing high-school teachers with Computer Science university-level studies are extremely important. For example, I have recently learned about the successful CS4HS Workshops organized by Carnegie-Mellon University (Carnegie-Mellon University, 2006 2008).

Actually, these lines were partly inspired by a similar (though at a smaller scale) initiative at my school, the Barcelona School of Informatics (`www.fib.upc.edu`). The school organized a meeting open to all secondary-education teachers, to exchange views and explore possible cooperations. As one of the points in the agenda, the school wanted to ask the teachers what could universities do to help them encouraging students to pursue computing studies. Can we provide teacher training? Can we offer disk space for file and webhosting? Should we visit high schools more often and give flashy talks? Send monthly bulletins via email?

The answer from the teachers was clear. Plainly expressed: "Well, all of this would be great. But if you really, really want to help us, please provide us with interesting materials, lesson plans, and activities. Ideally, it's stuff we can download, unpack, and bring to the classroom essentially as-it-is. We spend a lot of our time preparing material for our regular courses. We have neither the time nor the knowledge to prepare materials and activities to introduce our students to Computer Science". I have heard similar comments several times. It is a still a small sample, and restricted to the Spanish context, but I have no reason to believe that the answer would be substantially different elsewhere.

3. Why Machine Learning is so Appropriate?

In my opinion, the Machine Learning community can help (therefore: *should* help) with the student enrolment crisis more than most other fields of Computer Science.

I believe it is possible to explain to high-school students that CS in general, and Machine Learning in particular, is a challenging, exciting field. In particular, that progress on the "big problems" mentioned above (understanding the cell, curing cancer, finding the black matter among galaxies) strictly depends today on cutting-edge CS and Machine Learning research. And that as time goes, the number of challenging application fields keeps growing. Certainly, not all computing related jobs are equally rewarding intellectually — but neither do all jobs that engineers, medical doctors, architects, or biologists get.

More in particular, not many subfields of CS have the following two features, which I think are true of Machine Learning: 1) It is possible to explain some of the foundations of the field with the mathematical knowledge that (at least in theory) high-school students have. And 2) there are applications, from simple ones to extremely challenging ones, that can be explained in rigorous but understandable ways.

Concerning 1), "the basic math behind ML can be explained", note that basic concepts of linear algebra, vectors and vector operations, probability, and statistics suffice to explain how simple predictive models and simple unsupervised learning methods work, as well as the notions of classification error, testing, cross-validation, etc. Teachers may in fact welcome this as a way of reinforcing the math learned elsewhere and showing that it is not completely useless.

Concerning 2), "the challenges can be explained", we could all produce examples where Machine Learning helps genetics, drug design, cancer research, medical diagnosis, remote homecare for aged people, domotics, astronomy, not to mention economics and business.

For comparison, consider how hard it would be to explain to high-school students the methods, goals, and possible benefits of, say, theory of computation; programming language design and semantics; supercomputing; VLSI circuit design; software engineering; operating system internals; networking protocols and standards; database management system implementation, and so on.

Robotics and Computer Graphics are two other subdisciplines of CS whose very basic math can be understood at the high-school level, and whose "outcome" students certainly find exciting. However, it is not easy to do much in Robotics and Computer Graphics without programming. In contrast, I believe that it is possible to design a good number of sessions and activities using pre-programmed Machine Learning tools that do not require the students to write any programs at all. Machine Learning therefore helps conveying the idea that Computer Science is not exclusively about writing programs.

4. What Machine Learning can be Taught in High School?

We arrive now to the complicated details: what exactly could be taught in such module or modules? Admittedly, these are only very preliminary thoughts – but I am very much willing to discuss any ideas.

The basic techniques that could be explained and illustrated include at least:

- k-nearest neighbors classifiers
- Basic linear classifiers (say, the perceptron and possibly winnow)
- Naive Bayes
- Simple clustering algorithms (k-means and agglomerative algorithms)
- Frequent set and association rule mining

Note again that only basic vector terminology, probability (up to Bayes' Rule) and the notion of implication are strictly necessary to understand how the methods work. Ideally, the principle behind each method is explained to the students, who can then experiment with it using a given implementation. Later, more advanced methods can be provided as black boxes, with less or no detail about their inner workings, to deal with larger or more complex problems.

A critical issue is to come up with the right examples (scenarios, challenges, datasets). The difficulty is that the problem should be realistic enough to interest the students and the same time be "easy" enough that they can obtain meaningful results in reasonable time, before they find the task boring or frustrating. Identifiying, preparing, and testing these examples would be one of the longest and most delicate stages of a possible "Machine Learning for Secondary Education" project.

5. Thoughts for a Project

In this section I collect a few (incomplete, disparate, randomly ordered) thoughts about a possible project aimed at developing teaching materials introducing aspects of Machine Learning in secondary education.

I specifically think of a project created as an initiative within the PASCAL2 NOE, probably within its Curriculum Development programme, with cooperation from other agents. PASCAL2 is probably the largest organized group of Machine Learning experts in Europe, so it is a natural source of scientific expertise for such a project. It has the critical mass and the sufficient breadth to help deploy and test the project in a significant number of contexts within Europe – even though probably a very limited number of PASCAL2 sites would be involved in the actual development phases. Experiences such as the Videolectures website in its predecessor PASCAL network have shown the tremendous potential of cooperation within the network.

Some of the obvious technical requisites of the materials produced would be:

- They certainly should be free (as in "free lunch"). Although, in principle, materials should be usable without students or teachers having to do any programming, it is just a matter of time until some who know how to program feel like extending or modifying the software, and therefore being open-source is essentially a must. A number of open-source machine learning packages exist which could, or not, be used as a basis.
- The software should be adaptable to a huge variety of contexts: hardware, operating systems, content / courseware management systems, e-learning platforms, etc.
- In particular, if the project is indeed intending to be tested in several countries, localization would be a major issue. Careful design should ensure that porting to new languages is easy. Localization to different educational systems (with different curricula, different teaching cultures, etc.) would be an important challenge.

As mentioned, such a project would require strong coordination and cooperation with other agents, including, at least, national and regional educational agencies, a fair number of high schools themselves, and other projects or entities that may be involved in the promotion of CS studies. The development of a network of cooperating high schools is critical for the design, dissemination, and validation of the teaching modules. Eventually, it is them who have to say whether the product is interesting, adequate for realistic high-school education, technologically and pedagogically manageable, etc.

It would of course be advisable to look for cooperation or co-coordination with the local educational authorities at the most active sites. On the one hand, their endorsement would probably encourage more schools to join in the experience (although, as I have expressed before, I am optimistic that the initiative would be welcome by many teachers anyway). On the other hand, they could probably facilitate the contact with the most appropriate schools, contacts that PASCAL2 researchers will not have in general. Finally, educational agencies may be aware of other initiatives aimed at promoting technical studies among high-school students — for example, one such initiative is being set up right now by the regional government of Catalonia, where the crisis seems to be particularly acute.

References

Carnegie-Mellon University (2006-2008). CS4HS: Explorations in computer science for high school teachers. `http://www.cs.cmu.edu/cs4hs/`.